



## **PostMachine™ v1.1**



Ronald Eggers [ronald.eggerts@gte.net](mailto:ronald.eggerts@gte.net)  
Qian Chang [qchang@dhvx20.csudh.edu](mailto:qchang@dhvx20.csudh.edu)  
Mando Gomez [mandolux@yahoo.com](mailto:mandolux@yahoo.com)

**Dr. C.R. Williams**  
**Math 361 - Finite Automata**  
**California State University Dominguez Hills**

**May, 2000**

# Post Machine: PostMachine v1.1 Licensing Agreement



Ronald Eggers [ronald.eggers@gte.net](mailto:ronald.eggers@gte.net)  
Qian Chang [qchang@dhvx20.csudh.edu](mailto:qchang@dhvx20.csudh.edu)  
Mando Gomez [mandolux@yahoo.com](mailto:mandolux@yahoo.com)

## **Don't mess with us, we *WILL* send the blood sucking lawyers after you....**

Congratulations in getting PostMachine version 1.1. We hope this program and documentation will help you understand what a Post Machine does and how it does it.

Now, you are obligated to follow and obey by the following rules:

- 1) © 2000 PostMachine Software Corporation. PostMachine, the PostMachine logo, and PostMachineDudes image are trademarks of PostMachine Corporation. Microsoft Windows and MS-DOS are register trademarks of Microsoft Corporation. Apple Macintosh (Mac) is a register trademark of Apple Computer, Inc. All other trademarks are property of their respective holders.
- 2) The software, documentation and any images accompanying this License whether on disk, in read only memory, on any other media or in any other form (the "PostMachine Software") are licensed to you by PostMachine Corporation. We own the media on whom the PostMachine Software is recorded and PostMachine Corporation retains title to the PostMachine Software. The PostMachine Software in this package and any copies that this license authorizes you to make are subject to this license.
- 3) PostMachine does not warrants the media on which the PostMachine Software is recorded to be free from defects and workmanship under normal use for a period of ninety minutes from the date of original retail purchase.
- 4) You expressly acknowledge and agree that use of the PostMachine Software is at your own risk. The PostMachine Software is provided "AS IS" and without warranty of any kind and PostMachine EXPRESSLY DISCLAIMS ALL WARRANTIES AND/OR CONDITIONS, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES AND/OR CONDITIONS OF MERCHANTABILITY OR SATISFACTORY QUALITY AND FITNESS PURPOSE AND NONFRINGEMENT OF THIRD PARTIES RIGHTS.
- 5) PostMachine DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED ON THE PostMachine SOFTWARE WILL MEET YOUR REQUIREMENTS, OR THAT THE OPERATION OF THE PostMachine SOFTWARE WILL BE UNINTERRUPTED OR ERROR-

FREE, OR THAT DEFECTS IN THE PostMachine SOFTWARE WILL BE CORRECTED. FURTHERMORE, PostMachine DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE OF THE RESULTS OF THE USE OF THE PostMachine SOFTWARE OR RELATED DOCUMENTATIONS IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY PostMachine SHALL CREATE OR IN ANY WAY INCREASES THE SCOPE OF THIS WARRANTY. SHOULD THE PostMachine SOFTWARE PROVE DEFECTIVE, YOU (AND NOT PostMachine) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. SOME JURISDICTION DOES NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

- 6) THE TERMS OF THIS DISCLAIMER DO NOT AFFECT OR PREJUDICE THE STATUTORY RIGHTS OF A CONSUMER ACQUIRING PostMachine PRODUCTS OTHERWISE THAN IN THE COURSE OF A BUSINESS, NEITHER DO THEY LIMIT OR EXCLUDE ANY LIABILITY FOR DEATH OR PERSONAL INJURY CAUSED BY PostMachine's NEGLIGENCE.
- 7) If the PostMachine Software is supplied to the United States Government, the PostMachine Software is classified as "restricted computer software" as defined in clause 52.227-19 of the FAR. The United States Government's rights to the PostMachine Software are as provided in clause 52.227-19 of the FAR.
- 8) Complete Agreement. This License constitutes the entire agreement the parties with respect to the use of the PostMachine Software and supersedes all prior or contemporaneous understandings regarding such subject matter.
- 9) You WILL NOT disseminate nor distribute this software packet without writing consent of Ronald Eggers, Qian Chang, and Mando Gomez.
- 10) You WILL NOT change anything from this package and distribute it without prior consent of Ronald Eggers, Qian Chang, and Mando Gomez.
- 11) If you want to buy PostMachine v1.1, please send 20 American green backs to Ronald Eggers, Qian Chang, and Mando Gomez bank accounts: 5791-3211, 2345-6290, 5678-1141, respectively.
- 12) You WILL NOT share this software package with any other students. Let them suffer too ;-).
- 13) You WILL NOT use this software and/or documentation package to do any harm to anybody, especially Ronald Eggers, Qian Chang, and most definitely -- Mando Gomez.
- 14) We retained the right to grade the work of this project and therefore, this project represents a letter grade of A+ in a scale from F to A, A being the highest, and/or a digit grade of 10 in a scale of 1 to 10.

- 15) Whomever is reading, reviewing, or working with PostMachine v1.1 and has in his/her power to grade Ronald Eggers, Qian Chang, and Mando Gomez on any subject matter, must give the above individual a letter grade of A+ or a grade 10 or more in that subject matter.
- 16) Although we do not expect you to use this software in any mission critical application, but if you manage to start a nuclear warhead missile to the republic of either Afghanistan, China, Iran, Iraq (Saddam's Palace), North Korea, or some other funny place using this piece of code PLEASE DO SUE US. You won't get any money out of us but we will really enjoy the resulting publicity. Kindly break the news to us in a *gentle* way otherwise we may die laughing.
- 17) If you sue us nice and proper we all may become rich from TV and film rights and product endorsements. ;-)

The PostMachine Software team:

Ronald Eggers  
Qian Chang  
Mando Gomez

And our legal team:

Oops... we don't have the space in this book to list them all. To put on perspective how many and how good they are, the lawyers representing the Tobacco Industry are represented by our blood-sucking-ready-at-any-second-at-all-times-mean-killing-machines (aka Terminators)!

# Post Machine: Preface



Ronald Eggers [ronald.eggerts@gte.net](mailto:ronald.eggerts@gte.net)  
Qian Chang [qchang@dhvx20.csudh.edu](mailto:qchang@dhvx20.csudh.edu)  
Mando Gomez [mandolux@yahoo.com](mailto:mandolux@yahoo.com)

## What's On This Book?

The purpose of this book is to demonstrate one of the most interesting of the many finite automata machines -- the Post machine. A Post machine is an abstract machine consisting of a set of states (including the initial state), a set of input events, a set of output events, a state transition function, and certain scheme for storage. A Post machine incorporates a circular queue for STORAGE and has a START, ACCEPT, REJECT, ADD, READ, and CONSULT STATES, and a WORD define by an ALPHABET. The machine takes these components the same way a Universal Turing Machine works and makes the decision by the given data to accept or reject the given input. A Universal Turing Machine or UTM is what computers are (*machines that can simulate other machines*).

The algorithm for the Post machine in this book was influenced by *Introduction To Computer Theory, Second Edition* by Daniel I. A. Cohen from John Wiley & Sons Publishing. PostMachine v1.1 is similar to your average Post machine but differs in the implementation of such machine. For a complete definition of PostMachine v1.1 look at CHAPTER 1.

Some of the contents of this book are:

- A short biography of Emil Leon Post, the creator of the Post machine
- A short intro what a Post machine is, what it does, and how it does it
- What is PostMachine v1.1
- Sample runs using PostMachine v1.1 (Appendix)
- Installing PostMachine v1.1 in a PC
- How to use PostMachine v1.1
- Analyst Notes of PostMachine v1.1
- FAQ's

We, at PostMachine Software Corporation, wish you a very pleasant experience with our product and we hope you will appreciate the hard work that went into the conception, creation, implementation, testing, quality assurance, and dog gone it, hard and sweaty work that went into it. We are proud of our work, but as in life, there is always room for improvement.... ;-)

Ronald Eggers

Qian Chang

Mando Gomez

# Post Machine: Table of Contents



Ronald Eggers [ronald.eggert@gte.net](mailto:ronald.eggert@gte.net)  
Qian Chang [qchang@dhvx20.csudh.edu](mailto:qchang@dhvx20.csudh.edu)  
Mando Gomez [mandolux@yahoo.com](mailto:mandolux@yahoo.com)

## CHAPTER 1: Introduction

Welcome to PostMachine v1.1, 7  
Emil Leon Post, 7  
What is a Post machine?, 8  
What does a Post Machine do and how it does it?, 9  
What is PostMachine v1.1?, 11

## CHAPTER 2: Installing PostMachine v.1.1

Installation and User Documentation, 21  
Software and Hardware Installation Requirements, 21  
Installing PostMachine v1.1, 22  
Help Installing PostMachine v1.1 in Your Computer, 23  
User Documentation, 24

## CHAPTER 3: Using PostMachine/First Run

Using PostMachine v.1.1 and First Run, 26  
Problems Using PostMachine v1.1, 40

## CHAPTER 4: Analyst Notes

4.1 The class diagram, 39  
4.2 Class descriptions, 40  
    4.2.1 PostMachine class, 40  
    4.2.2 CharQueue class, 46  
    4.2.3 TransitionTable class, 49  
    4.2.4 TransitionTableField class, 54  
    4.2.5 Alphabet class, 60  
    4.2.6 AlphabetField class, 62

## CHAPTER 5: Frequently Asked Questions, 65

## APPENDIX, A-1

PostMachine v1.1 Source Code, A-67  
More Sample Runs Using PostMachine v1.1 , A-81  
     $a^n b^n$ , A-82  
     $a^n b^n_01$ , A-89  
     $a^n b^n c^n$ , A-96  
    notaaa, A-104  
    Palindromes, A-128  
    Palindromes Continuation, A-141

...

# Post Machine: CHAPTER 1



Ronald Eggers [ronald.eggert@gte.net](mailto:ronald.eggert@gte.net)  
Qian Chang [qchang@dhvx20.csudh.edu](mailto:qchang@dhvx20.csudh.edu)  
Mando Gomez [mandolux@yahoo.com](mailto:mandolux@yahoo.com)

---

## Welcome to PostMachine v1.1

### Emil Leon Post

Why are Post Machines called 'Post' Machines? Because they are called after the great mathematician Emil Leon Post (1897 - 1954) whose influence on computer science is particularly remarkable considering his lack of any contact with computing machines. Post approach was to study mathematical logical systems methods, which led him to conceptualize a machine that could simulate other machines.

Post moved to the United States at the age of seven with his family in 1904 from Augustów, Poland. The first subject which attracted him was astronomy. He attended the College of the City of New York receiving his B.A. in 1917. While at College he wrote his first paper, although it was not published for many years, on generalised differentiation. In 1917 he began research at Columbia University receiving his A.M in 1918 and his Ph.D. in 1920. In his Ph.D. thesis he proved the consistency of the propositional calculus described in Whitehead and Russell's *Principia Mathematica*. His work marks the beginning of proof theory.

After receiving his doctorate, Post went to Princeton for a year as Proctor Fellow. He returned to Columbia and, shortly after this, he had his first bout of an illness which was to recur throughout his career and limit what he might have achieved. In 1924 Post went to Cornell but again became ill. He resumed work as a high school teacher in New York in 1927. Then in 1932 he was appointed to the City College. He left after a short spell but returned three years later and spent the rest of his life there.

Post introduced the concept of completeness and consistency in a paper on truth-table methods. He attributed these methods to CJ Keyser rather than to Charles Peirce and E. Schröder as had been done previously. In 1920's Post attained results similar to Gödel, Church and Turing but he did not publish them. He also made a mathematical study of Łukasiewicz' three-valued logic. At around this time he wrote in his diary: - *I study Mathematics as a product of the human mind not as absolute.*

In 1936 he proposed what is now known as a Post machine, a kind of automaton which predates the notion of a program which von Neumann studied in 1946. In 1941 he wrote:- *...mathematical thinking is, and must be, essentially creative...* but he said there are limitations and symbolic logic is *...the indisputable means for revealing and developing these limitations.*

Post showed that the word problem for semigroups was recursively insoluble in 1947, giving the solution to a problem posed by Thue in 1914. Quine, in a letter written in 1954 after Post's death, said: - *Modern proof theory, and likewise the modern theory of machine computation, hinge on the concept of the recursive function. This important number theoretic concept... was discovered independently ... by four mathematicians, and one of these was Post. Subsequent work by Post was instrumental to the further progress of the theory of recursive functions.* Quine added in 1972:- *The theory of recursive functions of which Post was cofounder is now nearly twice as old as when I wrote that letter. What a fertile field it has proved to be.*

Paul Chessin recalls being taught by Post in New York in about 1943:- *I recall that he was a short stocky fellow who invariably dressed in a three piece suit, empty sleeve carefully tucked into the side suitcoat pocket. He would stride steadily up and down before the blackboard, speaking clearly, vigorous in his motions. He would frequently, suddenly whirl around to face the board, chalk in hand, to write. This motion always tended to loosen that sleeve from its anchor until finally (to the relief of the class) it flapped loosely about as a cape might. That freedom of motion seemed to us to liberate his thinking as he lectured.* Source: The MacTutor History of Mathematics Archive, <http://www-history.mcs.st-and.ac.uk/~history/>

## What is a Post machine?<sup>1</sup>

A Post machine is an abstract machine consisting of a set of states (including the initial state), a set of input events, a set of output events, and a state transition function.

What is an abstract machine?

- 1) A processor design which is not intended to be implemented as hardware, but which is the notional executor of a particular intermediate language (abstract machine language) used in a compiler or interpreter. An abstract machine has an instruction set, a register set and a model of memory. It may provide instructions which are closer to the language being compiled than any

---

<sup>1</sup> FOLDOC - Free On-line Dictionary of Computing, by Denis Howe. <http://burks.bton.ac.uk/burks/foldoc/>



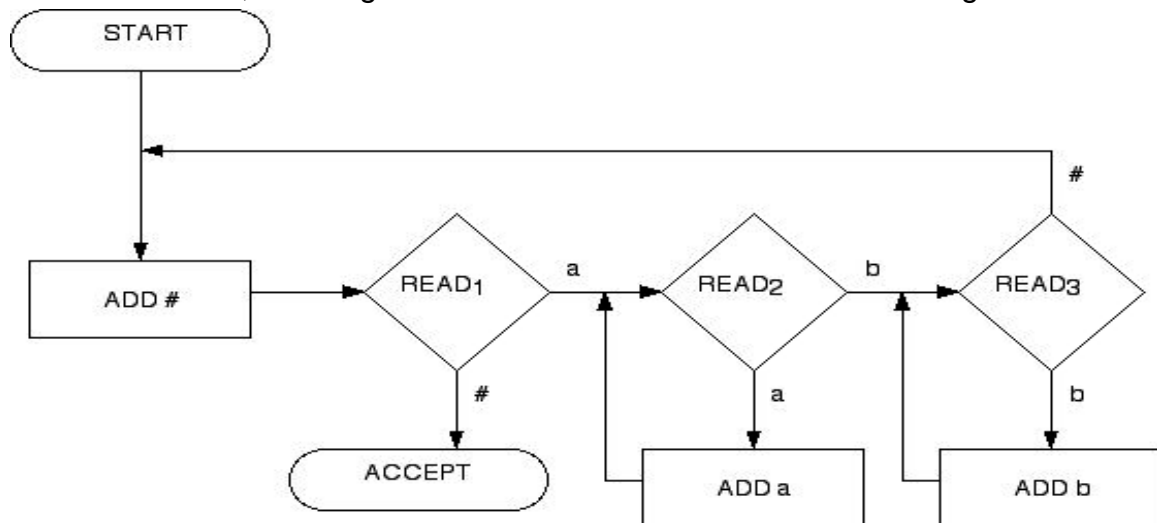
physical computer or it may be used to make the language implementation easier to port to other platforms.

- 2) A procedure for executing a set of instructions in some formal language, possibly also taking in input data and producing output. Such abstract machines are not intended to be constructed as hardware but are used in thought experiments about computability.

'So!' You say. Well, you may wander:

### What does a Post machine do? How does it do it?

A Post machine is a machine that incorporates a START, STORE, ACCEPT, REJECT, READ, and ADD STATES given a WORD define by an ALPHABET and it analyzes that WORD by the given STATES inputted into the machine (consult for example). The START STATE is automatically done at the start of the program. The STORE aspect of PostMachine v1.1 is a circular Abstract Data Type (ADT) queues where the given data is maintain in a loop. The queue takes care of incorporating READING and ADDing in a way that it always keeps the data in line defined by the user. A WORD is a concatenation of the elements of the given ALPHABET. For a simple example, lets examine the WORD of the form  $\{a^n b^n\}$ , 'aaabbb' whose ALPHABET is of course is 'a' and 'b'. To make life easier, lets diagram how the Post machine will do 'its thing.'



Source: *Introduction To Computer Theory, Second Edition* by Daniel I. A. Cohen from John Wiley & Sons Publishing. Page 459.

The Post machine STARTs evaluating the given WORD by putting it in STORE. Afterwards, it adds the '#' character (this character is

like a flag for the machine) to the WORD and again, it is put in the queue. As you can see, there is no explicit REJECT STATE drawn in the picture. Why? Because it is implied that when a letter is read and it has no edge leading out of the READ STATE, it is REJECTEd and the machine ceases.

The first 'a' is READ, then it is move to the second READ STATE which as you can observe, it goes into a loop. This 'loop' ADDs and READs all the 'a's that were in STORE (after reading the first 'a'). The net effect of this action is that the machine places one less 'a' onto the ass of the queue than there were originally in the WORD. This action is repeated for the 'b's. After all 'a's and 'b's are read and we come to read the '#' character we inputted in the beginning of this run, the machine is told to go back and add the # character onto the queue. All of the sudden, we are back from where we started but with one less set of the elements of the ALPHABET (a and b) in the queue. This action of removing a set of 'a and b' will in effect leaves us with a single '#' character in STORE and when it runs again, the machine will effect ACCEPT this word since that was one of the specifications of the diagram (WORD).

Now, in case you a doubter, this is part of *what* a Post machine can vomit, a trace of what happed in the course of examining the given data and utilizing its ability to STORE:

Enter word or 'N' for null string or '\*' to turn off machine: aaabbb

State = 1	START	Tape = aaabbb	Store = NULL
State = 2	READ	Tape = aabbb	Store = NULL
State = 3	STORE	Tape = aabbb	Store = A
State = 4	READ	Tape = abbb	Store = A
State = 3	STORE	Tape = abbb	Store = AA
State = 4	READ	Tape = bbb	Store = AA
State = 3	STORE	Tape = bbb	Store = AAA
State = 4	READ	Tape = bb	Store = AAA
State = 5	CONSULT	Tape = bb	Store = AA
State = 6	READ	Tape = b	Store = AA
State = 5	CONSULT	Tape = b	Store = A
State = 6	READ	Tape = NULL	Store = A
State = 5	CONSULT	Tape = NULL	Store = NULL
State = 6	READ	Tape = NULL	Store = NULL
State = 7	CONSULT	Tape = NULL	Store = NULL
State = 8	ACCEPT		

For the transition graph:

Number	State	Add	Queue	Next	If a	If b	If NULL	If A	If B	If NULL
0	X	-	-	0	0	0	0	0	0	0
1	I	-	-	2	0	0	0	0	0	0
2	R	-	-	0	3	0	8	0	0	0
3	W	A	A	4	0	0	0	0	0	0
4	R	-	-	0	3	5	0	0	0	0
5	K	-	-	0	0	0	0	6	0	0
6	R	-	-	0	0	5	7	0	0	0
7	K	-	-	0	0	0	0	0	0	8
8	A	-	-	0	0	0	0	0	0	0

'WHAT! That's IT! I want more examples...' You said. Well, we got what you want in the APPENDIX where you'll *enjoy*:  $a^n b^n$ ,  $a^n b^n a^n$ ,  $a^n b^n_{ab}$ , notaaa, and PALINDROMES. In the next section (below), you'll see the complete run of aaabbbb using PostMachine v1.1.

## What is PostMachine v1.1?

PostMachine v1.1 processes a given ALPHABET and TRANSITION table. The purpose of this program(s) is to drive and simulate a Post machine description (program). The user (you) will describe a given Post machine. When prompted, the user will provide its ALPHABET and its TRANSITION STATES (TRANSITION table) of the machine. The user will be prompted to define a Post machine as follows:

- 1) User will provide an ALPHABET and its size
- 2) User will provide the number of STATES of the machine description exclusively the number of REJECT STATES. The states of START, READ, CONSULT, STORE, SHIFT RIGHT CYCLICALLY (SRC), and ADD FRONT (AF) are to be counted. By protocol, the user will designate a REJECT state as an integer value of zero. This will be identified to the first row of the TRANSITION table with its array index as zero.
- 3) By protocol, all simulations will begin at a START STATE. This will be identified as the second row of the TRANSITION table with its array indexes as one with a STATE 1: START prompt, the user enters the NEXT STATE to TRANSITION.
- 4) FROM STATE 2 to STATE  $n$ , the user will select a STATE TYPE by entering a menu selection of available STATES. The user enters a corresponding number to make this selection.

This is as follows:

```

1:Counsult Queue (Store)
2:Store Queue (Store)
3:Shift Right Cyclically (SRC)

```

4:Add Front (AF)  
5:Accept

Once this selection is made. The user supplies additional information in context of the given STATE TYPE. Once the ALPHABET and its TRANSITION table description is completed. The user can now test the machine description as follows:

- 1) Enter a word with one or more letters of the given alphabet
- 2) Enter 'N' for the null string, or
- 3) Enter '\*' for turn-off the machine

Each entry will provide a trace of its Post machine execution from its initial state of START to its final state of ACCEPT or REJECT. There may be situations of infinite loops. The protocol calls for it to be an arbitrary number of TRANSITIONS through the machine. This may indicate that there may be an infinite loop. After a certain number of iterations, the machine will stop processing; it will prompt the user to test another WORD against the machine.

If the input WORD is the same as aaabbb, this is what you get with PostMachine v1.1 (and then some...) :

Menu:

1: Run Post Machine  
2: Quit

Enter: 1

Enter Tape alphabet:

Enter the size of the alphabet: 2  
Enter letter: a  
Enter letter: b

Alphabet:  
Letter 1: a  
Letter 2: b

Enter Store alphabet:

Enter the size of the alphabet: 2  
Enter letter: A  
Enter letter: B

Alphabet:  
Letter 1: A  
Letter 2: B

Defining Post Machine:

All Reject States are defined as 0 (zero).

Enter the number of states excluding rejected states: 8

State 1: START

Enter the next state (integer) : 2

State 2:

State Type:

- 1: Read Tape
- 2: Consult Queue
- 3: Store Queue
- 4: Shift Right Cyclically (SRC)
- 5: Add Front (AF)
- 6: Accept State

Enter a number: 1

Enter Next State (integer) :

- If a: 3
- If b: 0
- If NULL: 8

State 3:

State Type:

- 1: Read Tape
- 2: Consult Queue
- 3: Store Queue
- 4: Shift Right Cyclically (SRC)
- 5: Add Front (AF)
- 6: Accept State

Enter a number: 3

Enter letter to add at end of queue: A

Enter the next state (integer) : 4

State 4:

State Type:

- 1: Read Tape
- 2: Consult Queue
- 3: Store Queue
- 4: Shift Right Cyclically (SRC)
- 5: Add Front (AF)
- 6: Accept State

Enter a number: 1

Enter Next State (integer) :

    If a: 3  
    If b: 5  
    If NULL: 0

State 5:

State Type:

1: Read Tape  
2: Consult Queue  
3: Store Queue  
4: Shift Right Cyclically (SRC)  
5: Add Front (AF)  
6: Accept State

Enter a number: 2

Enter Next State (integer) :

    If A: 6  
    If B: 0  
    If NULL: 0

State 6:

State Type:

1: Read Tape  
2: Consult Queue  
3: Store Queue  
4: Shift Right Cyclically (SRC)  
5: Add Front (AF)  
6: Accept State

Enter a number: 1

Enter Next State (integer) :

    If a: 0  
    If b: 5  
    If NULL: 7

State 7:

State Type:

1: Read Tape  
2: Consult Queue  
3: Store Queue  
4: Shift Right Cyclically (SRC)  
5: Add Front (AF)  
6: Accept State

Enter a number: 2

Enter Next State (integer) :

If A: 0

If B: 0

If NULL: 8

State 8:

State Type:

1: Read Tape

2: Consult Queue

3: Store Queue

4: Shift Right Cyclically (SRC)

5: Add Front (AF)

6: Accept State

Enter a number: 6

Print Transition Table:

Number	State	Add	Queue	Next	If a	If b	If NULL	If A	If B	If
NULL										
0	X	-		0	0	0	0	0	0	0
1	I	-		2	0	0	0	0	0	0
2	R	-		0	3	0	8	0	0	0
3	W	A		4	0	0	0	0	0	0
4	R	-		0	3	5	0	0	0	0
5	K	-		0	0	0	0	6	0	0
6	R	-		0	0	5	7	0	0	0
7	K	-		0	0	0	0	0	0	8
8	A	-		0	0	0	0	0	0	0

Legend for State Type:

I = Start, A = Accept, X=Reject

R = Read, K = Consult, W = Store,

S=Shift Right Cyclically, F = Add Front

Enter word or 'N' for null string or '\*' to turn off machine: N

State = 1 START Tape = NULL Store = NULL

State = 2 READ Tape = NULL Store = NULL

State = 8 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: a

State = 1 START Tape = a Store = NULL

State = 2 READ Tape = NULL Store = NULL

State = 3 STORE Tape = NULL Store = A

State = 4 READ Tape = NULL Store = A

State = 0 REJECT

Enter word or 'N' for null string or '\*' to turn off machine: b

State =	1	START	Tape =	b	Store =	NULL
State =	2	READ	Tape =	NULL	Store =	NULL
State =	0	REJECT				

Enter word or 'N' for null string or '\*' to turn off machine: ab

State =	1	START	Tape =	ab	Store =	NULL
State =	2	READ	Tape =	b	Store =	NULL
State =	3	STORE	Tape =	b	Store =	A
State =	4	READ	Tape =	NULL	Store =	A
State =	5	CONSULT	Tape =	NULL	Store =	NULL
State =	6	READ	Tape =	NULL	Store =	NULL
State =	7	CONSULT	Tape =	NULL	Store =	NULL
State =	8	ACCEPT				

Enter word or 'N' for null string or '\*' to turn off machine: aab

State =	1	START	Tape =	aab	Store =	NULL
State =	2	READ	Tape =	ab	Store =	NULL
State =	3	STORE	Tape =	ab	Store =	A
State =	4	READ	Tape =	b	Store =	A
State =	3	STORE	Tape =	b	Store =	AA
State =	4	READ	Tape =	NULL	Store =	AA
State =	5	CONSULT	Tape =	NULL	Store =	A
State =	6	READ	Tape =	NULL	Store =	A
State =	7	CONSULT	Tape =	NULL	Store =	NULL
State =	0	REJECT				

Enter word or 'N' for null string or '\*' to turn off machine:  
aabbb

State =	1	START	Tape =	aabbb	Store =	NULL
State =	2	READ	Tape =	abbb	Store =	NULL
State =	3	STORE	Tape =	abbb	Store =	A
State =	4	READ	Tape =	bbb	Store =	A
State =	3	STORE	Tape =	bbb	Store =	AA
State =	4	READ	Tape =	bb	Store =	AA
State =	5	CONSULT	Tape =	bb	Store =	A
State =	6	READ	Tape =	b	Store =	A
State =	5	CONSULT	Tape =	b	Store =	NULL
State =	6	READ	Tape =	NULL	Store =	NULL
State =	5	CONSULT	Tape =	NULL	Store =	NULL
State =	0	REJECT				

Enter word or 'N' for null string or '\*' to turn off machine:  
aaabbb

State =	1	START	Tape =	aaabbb	Store =	NULL
State =	2	READ	Tape =	aaabbb	Store =	NULL
State =	3	STORE	Tape =	aaabbb	Store =	A
State =	4	READ	Tape =	abbb	Store =	A



State = 3	STORE	Tape = abbb	Store = AA
State = 4	READ	Tape = bbb	Store = AA
State = 3	STORE	Tape = bbb	Store = AAA
State = 4	READ	Tape = bb	Store = AAA
State = 5	CONSULT	Tape = bb	Store = AA
State = 6	READ	Tape = b	Store = AA
State = 5	CONSULT	Tape = b	Store = A
State = 6	READ	Tape = NULL	Store = A
State = 5	CONSULT	Tape = NULL	Store = NULL
State = 6	READ	Tape = NULL	Store = NULL
State = 7	CONSULT	Tape = NULL	Store = NULL
State = 8	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine:  
aaabbbbaa

State = 1	START	Tape = aaabbbbaa	Store = NULL
State = 2	READ	Tape = aabbbbaa	Store = NULL
State = 3	STORE	Tape = aabbbbaa	Store = A
State = 4	READ	Tape = abbbbaa	Store = A
State = 3	STORE	Tape = abbbbaa	Store = AA
State = 4	READ	Tape = bbbbaa	Store = AA
State = 3	STORE	Tape = bbbbaa	Store = AAA
State = 4	READ	Tape = bbaa	Store = AAA
State = 5	CONSULT	Tape = bbaa	Store = AA
State = 6	READ	Tape = baa	Store = AA
State = 5	CONSULT	Tape = baa	Store = A
State = 6	READ	Tape = aa	Store = A
State = 5	CONSULT	Tape = aa	Store = NULL
State = 6	READ	Tape = a	Store = NULL
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine:  
abbbaabb

State = 1	START	Tape = abbbaabb	Store = NULL
State = 2	READ	Tape = bbaaabb	Store = NULL
State = 3	STORE	Tape = bbaaabb	Store = A
State = 4	READ	Tape = baaabb	Store = A
State = 5	CONSULT	Tape = baaabb	Store = NULL
State = 6	READ	Tape = aaabb	Store = NULL
State = 5	CONSULT	Tape = aaabb	Store = NULL
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine:  
aaaaaabbabbb

State = 1	START	Tape = aaaaaabbabbb	Store = NULL
State = 2	READ	Tape = aaaaabbabbb	Store = NULL
State = 3	STORE	Tape = aaaaabbabbb	Store = A
State = 4	READ	Tape = aaaabbabbb	Store = A
State = 3	STORE	Tape = aaaabbabbb	Store = AA
State = 4	READ	Tape = aaabbabbb	Store = AA

State = 3	STORE	Tape = aaabbbbbbb	Store = AAA
State = 4	READ	Tape = aabbbbbbb	Store = AAA
State = 3	STORE	Tape = aabbbbbbb	Store = AAAA
State = 4	READ	Tape = abbbbbbb	Store = AAAA
State = 3	STORE	Tape = abbbbbbb	Store = AAAAA
State = 4	READ	Tape = bbbbbbb	Store = AAAAA
State = 3	STORE	Tape = bbbbbbb	Store = AAAAAA
State = 4	READ	Tape = bbbbbb	Store = AAAAAA
State = 5	CONSULT	Tape = bbbbbb	Store = AAAAA
State = 6	READ	Tape = bbbb	Store = AAAAA
State = 5	CONSULT	Tape = bbbb	Store = AAAA
State = 6	READ	Tape = bbb	Store = AAAA
State = 5	CONSULT	Tape = bbb	Store = AAA
State = 6	READ	Tape = bb	Store = AAA
State = 5	CONSULT	Tape = bb	Store = AA
State = 6	READ	Tape = b	Store = AA
State = 5	CONSULT	Tape = b	Store = A
State = 6	READ	Tape = NULL	Store = A
State = 5	CONSULT	Tape = NULL	Store = NULL
State = 6	READ	Tape = NULL	Store = NULL
State = 7	CONSULT	Tape = NULL	Store = NULL
State = 8	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine:  
abababbbbaa

State = 1	START	Tape = abababbbbaa	Store = NULL
State = 2	READ	Tape = bababbbbaa	Store = NULL
State = 3	STORE	Tape = bababbbbaa	Store = A
State = 4	READ	Tape = ababbbbaa	Store = A
State = 5	CONSULT	Tape = ababbbbaa	Store = NULL
State = 6	READ	Tape = babbbbaa	Store = NULL
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine:  
aaaaaaaaabbbbbbbbbb

State = 1	START	Tape = aaaaaaaaaabbbbbbbbbb	Store = NULL
State = 2	READ	Tape = aaaaaaaaaabbbbbbbbbb	Store = NULL
State = 3	STORE	Tape = aaaaaaaaaabbbbbbbbbb	Store = A
State = 4	READ	Tape = aaaaaaaaaabbbbbbbbbb	Store = A
State = 3	STORE	Tape = aaaaaaaaaabbbbbbbbbb	Store = AA
State = 4	READ	Tape = aaaaaaaaaabbbbbbbbbb	Store = AA
State = 3	STORE	Tape = aaaaaaaaaabbbbbbbbbb	Store = AAA
State = 4	READ	Tape = aaaaaaabbbbbbbbbb	Store = AAA
State = 3	STORE	Tape = aaaaaaabbbbbbbbbb	Store = AAAA
State = 4	READ	Tape = aaaaabbbbbbbbbb	Store = AAAA
State = 3	STORE	Tape = aaaaabbbbbbbbbb	Store = AAAAA
State = 4	READ	Tape = aaaabbbbbbbbbb	Store = AAAAA
State = 3	STORE	Tape = aaaabbbbbbbbbb	Store = AAAAAA
State = 4	READ	Tape = aaabbbbbbbbbb	Store = AAAAAA
State = 3	STORE	Tape = aaabbbbbbbbbb	Store = AAAAAA
State = 4	READ	Tape = aabbbbbbbbbb	Store = AAAAAA

```
Enter word or 'N' for null string or '*' to turn off machine:
aaaabbbbba
```

19

Enter word or 'N' for null string or '\*' to turn off machine:  
bbbaaa

State = 1 START           Tape = bbbaaa       Store = NULL  
State = 2 READ           Tape = bbaaa       Store = NULL  
State = 0 REJECT  
...want more? See the Appendix...

## Post Machine: CHAPTER 2



Ronald Eggers [ronald.eggerts@gte.net](mailto:ronald.eggerts@gte.net)  
Qian Chang [qchang@dhvx20.csudh.edu](mailto:qchang@dhvx20.csudh.edu)  
Mando Gomez [mandolux@yahoo.com](mailto:mandolux@yahoo.com)

---

### Installation and User Documentation

#### Software Requirements

- Any system that can use Disk Operating System (DOS) or MS-DOS. Windows 9x, NT, or 2000 should be able to utilize PostMachine v1.1.
- 4) Microsoft Office to review the documentation on this project. You can also utilize StarOffice from Sun Microsystems to read the .doc files (<http://www.sun.com/StarOffice>).
- 5) If you do not have Microsoft Office or StarOffice install in your computer, you can review the documentation by a web browser such as Netscape Communicator or Microsoft Internet Explorer. You can also examine the documentation on PostMachine v1.1 on the web at the following Internet address: <http://csc208.csudh.edu/agomez/Math361>.
- 6) If you have taste, you can review our documentation on Portable Document Format (PDF).

#### Hardware Requirements

- 7) Any computers with a CPU rated over 250Mhz, 32 MB of RAM.
- 8) A floppy disk drive.
- 9) At least 1 MB of disk space if you decide to run PostMachine v1.1 in your computer. Remember that you can run PostMachine v1.1 in the enclosed floppy disk. If

## Installing PostMachine v.1.1

- Read the label enclosing the brown envelop:

"The program on the enclosed disk(s) is licensed to the user.  
By opening this packet, you indicate your acceptance of the  
PostMachine Licensing Agreement"

The contents of that brown envelope will be the following:

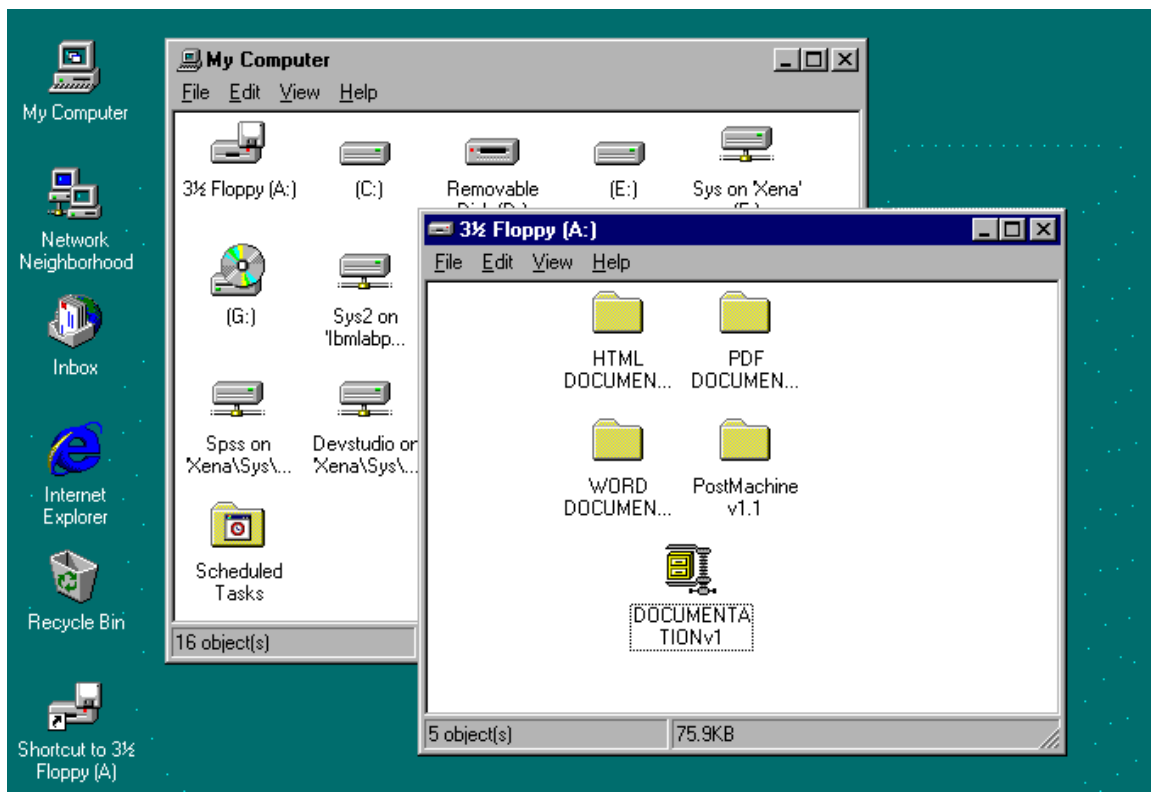
The License Agreement

A folder with the documentation for PostMachine v1.1

A folder with the source code of PostMachine v1.1

A white envelope containing the software media

- Open the white envelope label 'PostMachine Floppy Version.'
- Insert the floppy disk in your floppy drive.
- Go to 'My Computer' and look for your floppy drive icon ('3 1/2 Floppy [A:]'). Double click on it.



- You will see something similar to the image above. The contents of the floppy disk are the three formats of documentation (html, pdf, and Word), a folder named PostMachine v1.1, and a file containing all the files in compress mode (.zip). To open the zip file, you can use any utility that decompresses files such as Aladdin StuffIt (<http://www.aladdinsys.com/stuffit>) or MindVision MindExpander (<http://www.mindvision.com/mindexpander>).
- Double click on the folder 'PostMachine v1.1'
- Double click on 'POSTMACHINE.EXE' and a DOS window will pop-up if you want to run PostMachine v1.1 from the floppy disk.
- If speed is your need, create a directory (a folder) in your C Drive and deposit the contents of this floppy disk into that directory.
- Next, make sure you copied all the files from the floppy disk into the new directory and follow the steps disclose above.
- You are now ready to run PostMachine v1.1 from your computer.

## Help Installing PostMachine in Your Computer

If you have problems installing PostMachine v1.1, there must be something really wrong with your computer.

As for bugs in the software, our extensive testing and comparisons schemes have prove the quality assurance for any system running PostMachine v1.1 as indicated in the Hardware and Software Requirements.

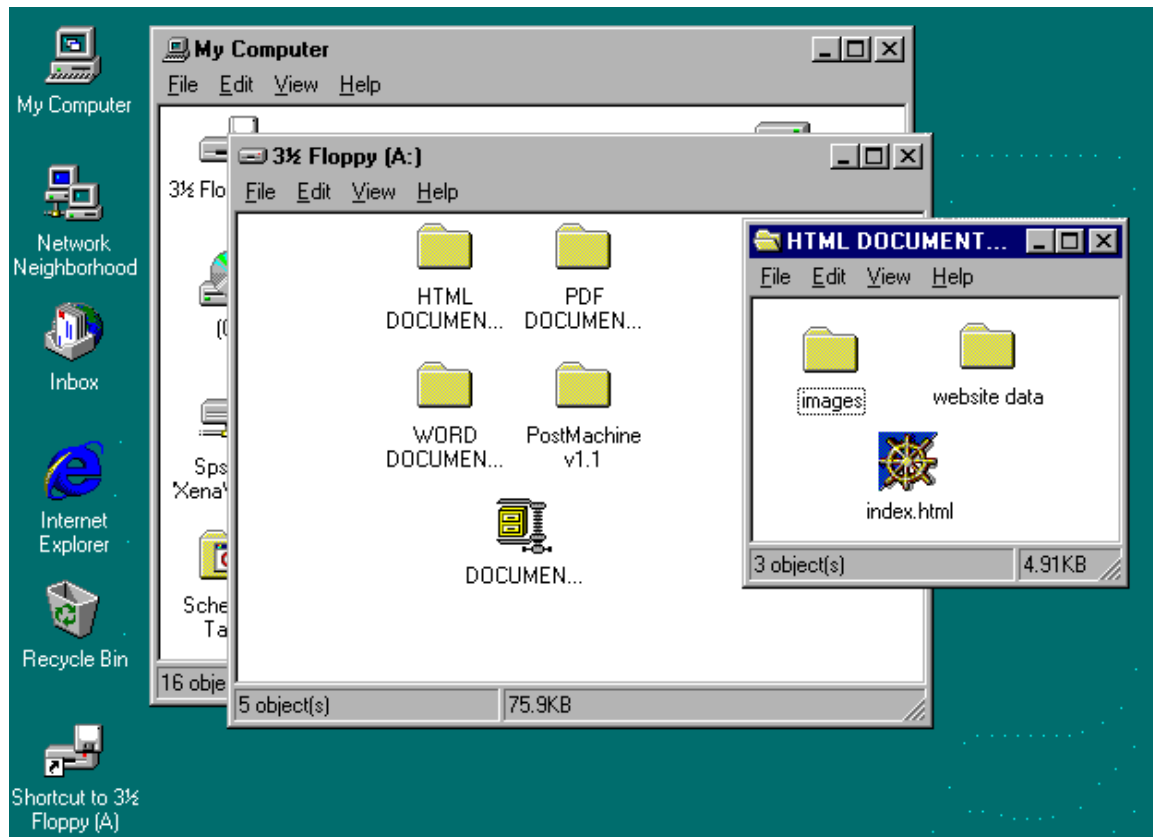
For customer service, please contact the PostMachine Development Team.

Ronald Eggers (the big kahuna), [ronald.eggers@gte.net](mailto:ronald.eggers@gte.net)  
 Qian Chang (the little kahuna), [qchang@dhvx20.csudh.edu](mailto:qchang@dhvx20.csudh.edu)  
 Mando Gomez (the dude), [mandolux@yahoo.com](mailto:mandolux@yahoo.com)

For the latest downloads, please visit  
<http://csc208.csudh.edu/agomez/Math361>

## User Documentation

- 1) Double click on the folder of your liking. Remember that there is HTML, PDF, and Word format of the documentation.
- 2) As you can see, if you open the 'HTML DOCUMENTATION' folder, you

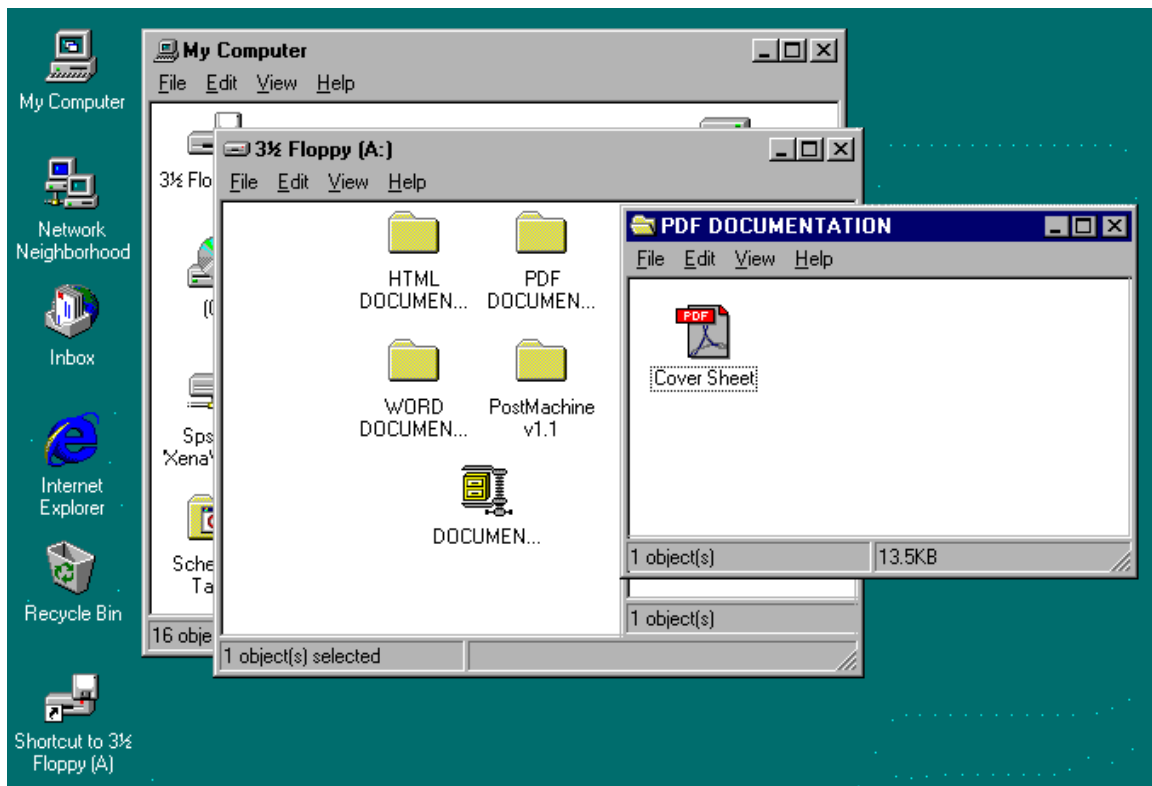


will see something similar to the following:

Double click on 'index.html' and your default web browser should come up and display a web page where you have access to all of the documentation and related objects for PostMachine v1.1. This is the mode of documentation we prefer for you to use.

- 3) If PDF is your format of choice (*style*), you will see a window with many PDF icons. Click on the icon of your choice.





- 4) If your choice is Word, you will see the same files as in the PDF folder except that you will be able to use them as you please (REMEMBER THE LICESE AGREEMENT... you don't want our team of blood sucker knocking at your front door, do you?).

## Post Machine: CHAPTER 3



Ronald Eggers [ronald.eggers@gte.net](mailto:ronald.eggers@gte.net)  
Qian Chang [qchang@dhvx20.csudh.edu](mailto:qchang@dhvx20.csudh.edu)  
Mando Gomez [mandolux@yahoo.com](mailto:mandolux@yahoo.com)

### Using PostMachine v1.1 - First Run

- Once you have fulfilled the requirements in CHAPTER 2, you can now play with PostMachine v1.1.
- Locate POSTMACHINE.EXE in your directory. Double click on it. And a DOS window will pop-up and it will begin to execute and run PostMachine v1.1.
- The program will show you a simple menu:

Menu:

```
1: Run Post Machine
2: Quit
```

Enter: \_

- Then it will ask you for the specifications of your ALPHABET. For example, let's enter an ALPHABET of 3 where we define the letters as a, b, and c (the next four lines are just to make sure you know what you inputted):

Enter Tape alphabet:

Enter the size of the alphabet: 3

Enter letter: a

Enter letter: b

Enter letter: c

Alphabet:

Letter 1: a

Letter 2: b

Letter 3: c

- Afterwards, the program will ask you for the 'Store alphabet:' what this is for the TRANSITION table for the program to utilize.
- After you input your ALPHABET, the program will start defining your Post machine and it will remind you that all 'All Reject States are defined as 0 (zero).'

- Now, you will be asked to input the number of states excluding rejected states you want.
- After defining the Post machine, the program will automatically go into START STATE.
- Afterwards, you will be asked to select the next STATE:

Enter the next state (integer) : 2

State 2:

State Type:

- 1: Read Tape
- 2: Consult Queue
- 3: Store Queue
- 4: Shift Right Cyclically (SRC)
- 5: Add Front (AF)
- 6: Accept State

Enter a number:

Now, its up to you to define your Post machine by selecting the NEXT STATE (the above represents our example using an ALPHABET of 3):

Enter Next State (integer) :

If a: \_  
If b: \_  
If c: \_  
If NULL: \_

- After your input all the STATE for the number of STATES you choose, say 9, we are ask again to select the 'State Type:' and if we choose to ACCEPT the given data, you will get a TRANSITION table.

State Type:

- 1: Read Tape
- 2: Consult Queue
- 3: Store Queue
- 4: Shift Right Cyclically (SRC)
- 5: Add Front (AF)
- 6: Accept State

Enter a number: 6

Number	State	Add Queue	Next	If a	If b	If c	If NULL	If A	If B	If NULL
0	X	-	0	0	0	0	0	0	0	0
1	I	-	2	0	0	0	0	0	0	0
2	R	-	0	3	4	0	0	0	0	0
3	W	A	2	0	0	0	0	0	0	0

4	W	B	5	3	5	0	0	0	0	0
5	K	-	0	0	0	0	0	6	0	0
6	R	-	0	0	4	7	8	0	0	0
7	K	-	0	0	0	0	0	0	6	0
8	K	-	0	0	0	0	0	0	0	9
9	A	-	0	0	0	0	0	0	0	0

Legend for State Type:

I = Start,      A = Accept,      X=Reject  
R = Read,        K = Consult,    W = Store,  
S=Shift Right Cyclically,    F = Add Front

- And finally, you can choose 'Enter word or 'N' for null string or '\*' to turn off machine:'

If you choose 'N', we get the following:

Enter word or 'N' for null string or '\*' to turn off machine: N

```
State = 1 START      Tape = NULL      Store = NULL
State = 2 READ       Tape = NULL      Store = NULL
State = 0 REJECT
```

If you choose to test a WORD with your already define machine, the program will create a trace of what was stored and define. If we continue with our example, if we input something like 'aaaaaaaabbbbbbbccccccc', we get the following monster:

Enter word or 'N' for null string or '\*' to turn off machine:  
aaaaaaaabbbbbbbccccccc

```
State = 1 START      Tape = aaaaaaabbbbbbbccccccc      Store = NULL
State = 2 READ       Tape = aaaaaaabbbbbbbccccccc      Store = NULL
State = 3 STORE      Tape = aaaaaaabbbbbbbccccccc      Store = A
State = 2 READ       Tape = aaaaaaabbbbbbbccccccc      Store = A
State = 3 STORE      Tape = aaaaaaabbbbbbbccccccc      Store = AA
State = 2 READ       Tape = aaaaaaabbbbbbbccccccc      Store = AA
State = 3 STORE      Tape = aaaaaaabbbbbbbccccccc      Store = AAA
State = 2 READ       Tape = aaaabbbbbbbccccccc         Store = AAA
State = 3 STORE      Tape = aaaabbbbbbbccccccc         Store = AAAA
State = 2 READ       Tape = aaabbbbbbbccccccc          Store = AAAA
State = 3 STORE      Tape = aaabbbbbbbccccccc          Store = AAAAA
State = 2 READ       Tape = aabbbbbbbccccccc           Store = AAAAA
State = 3 STORE      Tape = aabbbbbbbccccccc           Store = AAAAAA
State = 2 READ       Tape = abbbbbbbccccccc            Store = AAAAAA
State = 3 STORE      Tape = abbbbbbbccccccc            Store = AAAAAAA
State = 2 READ       Tape = bbbbbbbccccccc             Store = AAAAAAA
State = 3 STORE      Tape = bbbbbbbccccccc             Store = AAAAAAAA
State = 2 READ       Tape = bbbbbbbccccccc             Store = AAAAAAAA
State = 4 STORE      Tape = bbbbbbbccccccc             Store = AAAAAAAB
```

State = 5 CONSULT	Tape = bbbbbbbcccccccc	Store = AAAAAAAB
State = 6 READ	Tape = bbbbbbbcccccccc	Store = AAAAAAAB
State = 4 STORE	Tape = bbbbbbbcccccccc	Store = AAAAAAABB
State = 5 CONSULT	Tape = bbbbbbbcccccccc	Store = AAAAAABB
State = 6 READ	Tape = bbbbbbbcccccccc	Store = AAAAAABB
State = 4 STORE	Tape = bbbbbbbcccccccc	Store = AAAAAABBB
State = 5 CONSULT	Tape = bbbbbbbcccccccc	Store = AAAAAABB
State = 6 READ	Tape = bbbbbbbcccccccc	Store = AAAAAABB
State = 4 STORE	Tape = bbbbbbbcccccccc	Store = AAAAAABBB
State = 5 CONSULT	Tape = bbbbbbbcccccccc	Store = AAAAABBB
State = 6 READ	Tape = bbbcccccccc	Store = AAAAABBB
State = 4 STORE	Tape = bbbcccccccc	Store = AAAAABBB
State = 5 CONSULT	Tape = bbbcccccccc	Store = AAABBBB
State = 6 READ	Tape = bbcccccccc	Store = AAABBBB
State = 4 STORE	Tape = bbcccccccc	Store = AAABBBB
State = 5 CONSULT	Tape = bbcccccccc	Store = AABBBBB
State = 6 READ	Tape = bcccccccc	Store = AABBBBB
State = 4 STORE	Tape = bcccccccc	Store = AABBBBB
State = 5 CONSULT	Tape = bcccccccc	Store = ABBBBB
State = 6 READ	Tape = cccccccc	Store = ABBBBB
State = 4 STORE	Tape = cccccccc	Store = ABBBBB
State = 5 CONSULT	Tape = cccccccc	Store = ABBBBB
State = 6 READ	Tape = cccccccc	Store = ABBBBB
State = 4 STORE	Tape = cccccccc	Store = ABBBBB
State = 5 CONSULT	Tape = cccccccc	Store = BBBBBB
State = 6 READ	Tape = ccccccc	Store = BBBBBB
State = 7 CONSULT	Tape = ccccccc	Store = BBBBBB
State = 6 READ	Tape = ccccccc	Store = BBBBBB
State = 7 CONSULT	Tape = ccccccc	Store = BBBBBB
State = 6 READ	Tape = ccccc	Store = BBBBB
State = 7 CONSULT	Tape = ccccc	Store = BBBBB
State = 6 READ	Tape = ccccc	Store = BBBBB
State = 7 CONSULT	Tape = ccccc	Store = BBBBB
State = 6 READ	Tape = cccc	Store = BBBB
State = 7 CONSULT	Tape = cccc	Store = BBBB
State = 6 READ	Tape = ccc	Store = BBBB
State = 7 CONSULT	Tape = ccc	Store = BBB
State = 6 READ	Tape = cc	Store = BBB
State = 7 CONSULT	Tape = cc	Store = BB
State = 6 READ	Tape = c	Store = BB
State = 7 CONSULT	Tape = c	Store = B
State = 6 READ	Tape = NULL	Store = B
State = 7 CONSULT	Tape = NULL	Store = NULL
State = 6 READ	Tape = NULL	Store = NULL
State = 8 CONSULT	Tape = NULL	Store = NULL
State = 9 ACCEPT		

This is what is expected since we are using an  $a^n b^n c^n$  word (all parameters are follow: all ALPHABET components are of size  $n$ ). What if we input a word of the form where  $n$  is not the same, such as  $aabbcccc$  or  $a^2b^2c^4$ . If such WORD is tested, we should automatically realize that this would produce a bad machine or REJECTED WORD. This is what we get:

Enter word or 'N' for null string or '\*' to turn off machine: aabbcccc

State = 1 START	Tape = aabbcccc	Store = NULL
State = 2 READ	Tape = abbcccc	Store = NULL
State = 3 STORE	Tape = abbcccc	Store = A
State = 2 READ	Tape = bbcccc	Store = A
State = 3 STORE	Tape = bbcccc	Store = AA

State = 2	READ	Tape = bcccc	Store = AA
State = 4	STORE	Tape = bcccc	Store = AAB
State = 5	CONSULT	Tape = bcccc	Store = AB
State = 6	READ	Tape = cccc	Store = AB
State = 4	STORE	Tape = cccc	Store = ABB
State = 5	CONSULT	Tape = cccc	Store = BB
State = 6	READ	Tape = ccc	Store = BB
State = 7	CONSULT	Tape = ccc	Store = B
State = 6	READ	Tape = cc	Store = B
State = 7	CONSULT	Tape = cc	Store = NULL
State = 6	READ	Tape = c	Store = NULL
State = 7	CONSULT	Tape = c	Store = NULL
State = 0	REJECT		

The following is the complete run using the ALPHABET a b c example:

Menu:

1: Run Post Machine  
2: Quit

Enter: 1

Enter Tape alphabet:

Enter the size of the alphabet: 3  
Enter letter: a  
Enter letter: b  
Enter letter: c

Alphabet:

Letter 1: a  
Letter 2: b  
Letter 3: c

Enter Store alphabet:

Enter the size of the alphabet: 2  
Enter letter: A  
Enter letter: B

Alphabet:

Letter 1: A  
Letter 2: B

Defining Post Machine:

All Reject States are defined as 0 (zero).

Enter the number of states excluding rejected states: 9

State 1: START

Enter the next state (integer) : 2

State 2:

```

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 1

Enter Next State (integer) :
    If a: 3
    If b: 4
    If c: 0
    If NULL: 0

State 3:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 3
Enter letter to add at end of queue: A

Enter the next state (integer) : 2

State 4:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 3
Enter letter to add at end of queue: B

Enter the next state (integer) : 5

State 5:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 2

```

```

Enter Next State (integer) :
    If A: 6
    If B: 0
    If NULL: 0

State 6:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 1

Enter Next State (integer) :
    If a: 0
    If b: 4
    If c: 7
    If NULL: 8

State 7:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 2

Enter Next State (integer) :
    If A: 0
    If B: 6
    If NULL: 0

State 8:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 2

Enter Next State (integer) :
    If A: 0
    If B: 0
    If NULL: 9

```



State 9:

State Type:

- 1: Read Tape
- 2: Consult Queue
- 3: Store Queue
- 4: Shift Right Cyclically (SRC)
- 5: Add Front (AF)
- 6: Accept State

Enter a number: 6

Number	State	Add Queue	Next	If a	If b	If c	If NULL	If A	If B	If NULL
0	X	-	0	0	0	0	0	0	0	0
1	I	-	2	0	0	0	0	0	0	0
2	R	-	0	3	4	0	0	0	0	0
3	W	A	2	0	0	0	0	0	0	0
4	W	B	5	3	5	0	0	0	0	0
5	K	-	0	0	0	0	0	6	0	0
6	R	-	0	0	4	7	8	0	0	0
7	K	-	0	0	0	0	0	0	6	0
8	K	-	0	0	0	0	0	0	0	9
9	A	-	0	0	0	0	0	0	0	0

Legend for State Type:

I = Start,      A = Accept,      X=Reject  
R = Read,      K = Consult,      W = Store,  
S=Shift Right Cyclically,      F = Add Front

Enter word or 'N' for null string or '\*' to turn off machine: N

State = 1 START      Tape = NULL      Store = NULL  
State = 2 READ      Tape = NULL      Store = NULL  
State = 0 REJECT

Enter word or 'N' for null string or '\*' to turn off machine: a

State = 1 START      Tape = a      Store = NULL  
State = 2 READ      Tape = NULL      Store = NULL  
State = 3 STORE      Tape = NULL      Store = A  
State = 2 READ      Tape = NULL      Store = A  
State = 0 REJECT

Enter word or 'N' for null string or '\*' to turn off machine: bac

State = 1 START      Tape = bac      Store = NULL  
State = 2 READ      Tape = ac      Store = NULL  
State = 4 STORE      Tape = ac      Store = B  
State = 5 CONSULT      Tape = ac      Store = NULL  
State = 0 REJECT

Enter word or 'N' for null string or '\*' to turn off machine: abc

State = 1 START      Tape = abc      Store = NULL  
State = 2 READ      Tape = bc      Store = NULL  
State = 3 STORE      Tape = bc      Store = A

State = 2	READ	Tape = c	Store = A
State = 4	STORE	Tape = c	Store = AB
State = 5	CONSULT	Tape = c	Store = B
State = 6	READ	Tape = NULL	Store = B
State = 7	CONSULT	Tape = NULL	Store = NULL
State = 6	READ	Tape = NULL	Store = NULL
State = 8	CONSULT	Tape = NULL	Store = NULL
State = 9	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: aabbbcc

State = 1	START	Tape = aabbbcc	Store = NULL
State = 2	READ	Tape = abbbcc	Store = NULL
State = 3	STORE	Tape = abbbcc	Store = A
State = 2	READ	Tape = bbbcc	Store = A
State = 3	STORE	Tape = bbbcc	Store = AA
State = 2	READ	Tape = bbcc	Store = AA
State = 4	STORE	Tape = bbcc	Store = AAB
State = 5	CONSULT	Tape = bbcc	Store = AB
State = 6	READ	Tape = bcc	Store = AB
State = 4	STORE	Tape = bcc	Store = ABB
State = 5	CONSULT	Tape = bcc	Store = BB
State = 6	READ	Tape = cc	Store = BB
State = 4	STORE	Tape = cc	Store = BBB
State = 5	CONSULT	Tape = cc	Store = BB
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: aaabbcccc

State = 1	START	Tape = aaabbcccc	Store = NULL
State = 2	READ	Tape = aabbcccc	Store = NULL
State = 3	STORE	Tape = aabbcccc	Store = A
State = 2	READ	Tape = abbcccc	Store = A
State = 3	STORE	Tape = abbcccc	Store = AA
State = 2	READ	Tape = bbcccc	Store = AA
State = 3	STORE	Tape = bbcccc	Store = AAA
State = 2	READ	Tape = bcccc	Store = AAA
State = 4	STORE	Tape = bcccc	Store = AAAB
State = 5	CONSULT	Tape = bcccc	Store = AAB
State = 6	READ	Tape = ccc	Store = AAB
State = 4	STORE	Tape = ccc	Store = AABB
State = 5	CONSULT	Tape = ccc	Store = ABB
State = 6	READ	Tape = cc	Store = ABB
State = 7	CONSULT	Tape = cc	Store = BB
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: aabbccccc

State = 1	START	Tape = aabbccccc	Store = NULL
State = 2	READ	Tape = abbccccc	Store = NULL
State = 3	STORE	Tape = abbccccc	Store = A
State = 2	READ	Tape = bbccccc	Store = A
State = 3	STORE	Tape = bbccccc	Store = AA
State = 2	READ	Tape = bccccc	Store = AA
State = 4	STORE	Tape = bccccc	Store = AAB
State = 5	CONSULT	Tape = bccccc	Store = AB
State = 6	READ	Tape = cccc	Store = AB
State = 4	STORE	Tape = cccc	Store = ABB

```

State = 5 CONSULT    Tape = cccc    Store = BB
State = 6 READ       Tape = ccc     Store = BB
State = 7 CONSULT    Tape = ccc     Store = B
State = 6 READ       Tape = cc      Store = B
State = 7 CONSULT    Tape = cc      Store = NULL
State = 6 READ       Tape = c       Store = NULL
State = 7 CONSULT    Tape = c       Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaccbb

```

State = 1 START      Tape = aaccbb    Store = NULL
State = 2 READ       Tape = accbb     Store = NULL
State = 3 STORE      Tape = accbb     Store = A
State = 2 READ       Tape = cbb      Store = A
State = 3 STORE      Tape = cbb      Store = AA
State = 2 READ       Tape = cbb      Store = AA
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbaacc

```

State = 1 START      Tape = bbaacc    Store = NULL
State = 2 READ       Tape = baacc     Store = NULL
State = 4 STORE      Tape = baacc     Store = B
State = 5 CONSULT    Tape = baacc     Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine:  
aaaabbbbbaaaa

```

State = 1 START      Tape = aaaabbbbbaaaa    Store = NULL
State = 2 READ       Tape = aaabbbbbaaaa    Store = NULL
State = 3 STORE      Tape = aaabbbbbaaaa    Store = A
State = 2 READ       Tape = aabbbbbaaaa     Store = A
State = 3 STORE      Tape = aabbbbbaaaa     Store = AA
State = 2 READ       Tape = abbbbbaaaa      Store = AA
State = 3 STORE      Tape = abbbbbaaaa      Store = AAA
State = 2 READ       Tape = bbbbbaaaa       Store = AAA
State = 3 STORE      Tape = bbbbbaaaa       Store = AAAA
State = 2 READ       Tape = bbbbaaaa        Store = AAAA
State = 4 STORE      Tape = bbbbaaaa        Store = AAAAB
State = 5 CONSULT    Tape = bbbbaaaa        Store = AAAB
State = 6 READ       Tape = bbbaaaa         Store = AAAB
State = 4 STORE      Tape = bbbaaaa         Store = AAABB
State = 5 CONSULT    Tape = bbbaaaa         Store = AABB
State = 6 READ       Tape = baaaa          Store = AABB
State = 4 STORE      Tape = baaaa          Store = AABBB
State = 5 CONSULT    Tape = baaaa          Store = AB BB
State = 6 READ       Tape = aaaa           Store = AB BB
State = 4 STORE      Tape = aaaa           Store = AB BBB
State = 5 CONSULT    Tape = aaaa           Store = BB BB
State = 6 READ       Tape = aaa            Store = BB BB
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine:  
aaaaabbbbcccc

```

State = 1 START      Tape = aaaaabbbbcccc    Store = NULL

```

State = 2 READ	Tape = aaaabbbbbcccccc	Store = NULL
State = 3 STORE	Tape = aaaabbbbbcccccc	Store = A
State = 2 READ	Tape = aaabbbbbcccccc	Store = A
State = 3 STORE	Tape = aaabbbbbcccccc	Store = AA
State = 2 READ	Tape = aabbbbbcccccc	Store = AA
State = 3 STORE	Tape = aabbbbbcccccc	Store = AAA
State = 2 READ	Tape = abbbbbcccccc	Store = AAA
State = 3 STORE	Tape = abbbbbcccccc	Store = AAAA
State = 2 READ	Tape = bbbbbcccccc	Store = AAAA
State = 3 STORE	Tape = bbbbbcccccc	Store = AAAAA
State = 2 READ	Tape = bbbbcccccc	Store = AAAAA
State = 4 STORE	Tape = bbbbcccccc	Store = AAAAAB
State = 5 CONSULT	Tape = bbbbcccccc	Store = AAAAB
State = 6 READ	Tape = bbbcccccc	Store = AAAAB
State = 4 STORE	Tape = bbbcccccc	Store = AAAABB
State = 5 CONSULT	Tape = bbbcccccc	Store = AAABB
State = 6 READ	Tape = bbcccccc	Store = AAABB
State = 4 STORE	Tape = bbcccccc	Store = AAABBB
State = 5 CONSULT	Tape = bbcccccc	Store = AABBB
State = 6 READ	Tape = bcccccc	Store = AABBB
State = 4 STORE	Tape = bcccccc	Store = AABBBB
State = 5 CONSULT	Tape = bcccccc	Store = ABBBB
State = 6 READ	Tape = ccccc	Store = ABBBB
State = 4 STORE	Tape = ccccc	Store = ABBBBB
State = 5 CONSULT	Tape = ccccc	Store = BBBBB
State = 6 READ	Tape = cccc	Store = BBBBB
State = 7 CONSULT	Tape = cccc	Store = BBBBB
State = 6 READ	Tape = ccc	Store = BBBBB
State = 7 CONSULT	Tape = ccc	Store = BBB
State = 6 READ	Tape = cc	Store = BBB
State = 7 CONSULT	Tape = cc	Store = BB
State = 6 READ	Tape = c	Store = BB
State = 7 CONSULT	Tape = c	Store = B
State = 6 READ	Tape = NULL	Store = B
State = 7 CONSULT	Tape = NULL	Store = NULL
State = 6 READ	Tape = NULL	Store = NULL
State = 8 CONSULT	Tape = NULL	Store = NULL
State = 9 ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine:  
 aaaaaaaabbbbbbbcccccccc

State = 1 START	Tape = aaaaaaaabbbbbbbcccccccc	Store = NULL
State = 2 READ	Tape = aaaaaaaabbbbbbbcccccccc	Store = NULL
State = 3 STORE	Tape = aaaaaaaabbbbbbbcccccccc	Store = A
State = 2 READ	Tape = aaaaaaaabbbbbbbcccccccc	Store = A
State = 3 STORE	Tape = aaaaaaaabbbbbbbcccccccc	Store = AA
State = 2 READ	Tape = aaaaaaaabbbbbbbcccccccc	Store = AA
State = 3 STORE	Tape = aaaaaaaabbbbbbbcccccccc	Store = AAA
State = 2 READ	Tape = aaaabbbbbbbcccccccc	Store = AAA
State = 3 STORE	Tape = aaaabbbbbbbcccccccc	Store = AAAA
State = 2 READ	Tape = aaabbbbbbbcccccccc	Store = AAAA
State = 3 STORE	Tape = aaabbbbbbbcccccccc	Store = AAAAA
State = 2 READ	Tape = aabbbbbbbcccccccc	Store = AAAAA
State = 3 STORE	Tape = aabbbbbbbcccccccc	Store = AAAAAA
State = 2 READ	Tape = abbbbbbbcccccccc	Store = AAAAAA
State = 3 STORE	Tape = abbbbbbbcccccccc	Store = AAAAAA
State = 2 READ	Tape = bbbbbbbcccccccc	Store = AAAAAA



```
State = 6 READ      Tape = NULL      Store = BB
State = 8 CONSULT   Tape = NULL      Store = B
State = 0 REJECT
```

Enter word or 'N' for null string or '\*' to turn off machine: aaccbb

```
State = 1 START      Tape = aaccbb      Store = NULL
State = 2 READ        Tape = accbb       Store = NULL
State = 3 STORE        Tape = accbb       Store = A
State = 2 READ        Tape = ccbb        Store = A
State = 3 STORE        Tape = ccbb        Store = AA
State = 2 READ        Tape = cbb         Store = AA
State = 0 REJECT
```

## Problems Using PostMachine v1.1

If you encounter problems running or using PostMachine v1.1, please call us at 1-800-DO WE CARE? Unlike Microsoft Corporation, our customer assistant program cost an unbelievable low \$0.0285 a second ( ;- ) ). Beat that you *Bastards of Satan!*

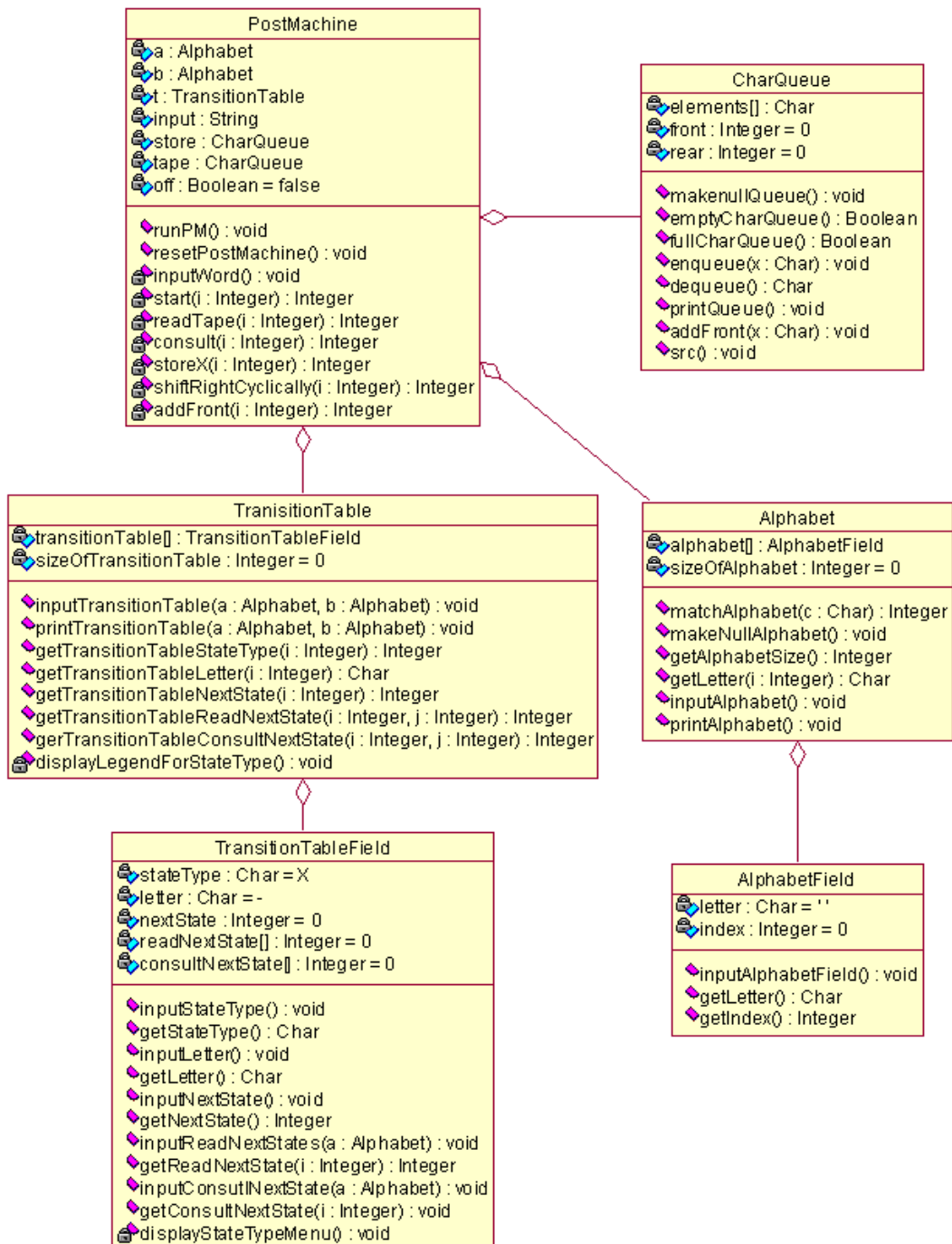
# Post Machine: CHAPTER 4



Ronald Eggers [ronald.eggert@gte.net](mailto:ronald.eggert@gte.net)  
Qian Chang [qchang@dhvx20.csudh.edu](mailto:qchang@dhvx20.csudh.edu)  
Mando Gomez [mandolux@yahoo.com](mailto:mandolux@yahoo.com)

## Analyst Notes

### 4.1 Class Diagram



## 4.2 Class Descriptions, Attributes and Operations

### 4.2.1 PostMachine class

#### 4.2.1.1 Description

The PostMachine class processes a given input string and a transition table. The purpose of the class is to drive and simulate a Post machine (program). The user will describe a given Post machine interactively. When prompted, the user will provide an input tape alphabet, a store alphabet if applicable, and a transition table of the Post machine.

The user will be prompted to define a Post machine as follows:

- (1) User will provide an input tape alphabet and its size.
- (2) If applicable, user will provide a store alphabet and its size.
- (3) User will provide the number of states of the machine description excluding the number of reject states. The states of Start, Read, Consult, Store, Shift Right Cyclically (SRC) and Add Front (AF) are to be counted. By protocol, the user will designate a reject state by an integer value of zero. This will be identified as the first row of the transition table with an array index of zero.
- (4) By protocol, all simulations will begin at a Start state. This will be identified as the second row of the transition table with its array index as one. With a 'State 1: Start' prompt, the user enters an integer for the next state of transition.
- (5) From State 2 to State n, the user will be presented with a menu to select one of the available state types. The user enters a corresponding number to make this selection.

This is as follows:

- 1: Read Tape
- 2: Consult Queue (Store)
- 3: Store Queue (Store)
- 4: Shift Right Cyclically (SRC)
- 5: Add Front (AF)
- 6: Accept

Once this selection is made. The user supplies additional information in context of the given State Type.

Once the input tape alphabet, store alphabet and transition table description is completed. The user can now test the machine description as follows:



- (1) Enter a word with one or more letters of the given alphabet,
- (2) Enter 'N' for the null string, or
- (3) Enter '\*' to turn off the machine.

Each entry will provide a trace of its Post machine execution from its initial state of Start to its final state of Accept or Reject. There may be situations of infinite loops. The protocol calls for it to be an arbitrary number of transitions through the machine. This may indicate that there may be an infinite loop. This is not proof of an infinite loop. After a certain number of iterations, the machine will stop processing. It will prompt the user to test another word against machine.

#### 4.2.1.2 PostMachine attributes

Name	Object Type	Access	Description
a	Alphabet	private	Input alphabet of Post machine
b	Alphabet	private	Store alphabet of Post machine
tape	CharQueue	private	Input tape of the Post machine
store	CharQueue	private	Store of the Post machine
t	TransitionTable	private	Transition table of Post machine
input	String	private	Input string of the Post machine

#### 4.2.1.3 PostMachine functions

##### 4.2.1.3.1 runPM()

<b>Name:</b>	runPM()
<b>Access:</b>	public
<b>Description:</b>	runPM() requires the user to input a input tape alphabet, a store alphabet, and a transition table. The user inputs a string to test a given machine description. The machine simulation tests a given string to determine if it is to be accepted or rejected or if a possible loop exists. To turn off the machine the user will enter '*' to stop the simulation.
<b>Return Type:</b>	void

<b>Parameters:</b>	void
<b>Pre-condition:</b>	The user decides to turn on the Post machine
<b>Post-condition:</b>	The user decides to turn off the Post machine
<b>Requires:</b>	The input tape alphabet is greater than zero and a transition table is defined.

#### 4.2.1.3.2 resetPostMachine

<b>Name:</b>	resetPostMachine()
<b>Access:</b>	public
<b>Description:</b>	resetPostMachine() resets the Post machine by setting off = false.
<b>Return Type:</b>	void
<b>Parameters:</b>	void
<b>Pre-condition:</b>	None
<b>Post-condition:</b>	Post machine is on
<b>Requires:</b>	None

#### 4.2.1.3.3 inputWord()

<b>Name:</b>	inputWord()
<b>Access:</b>	private
<b>Description:</b>	<p>inputWord() prompts the user to enter a word to test the Post machine description. The user's input is stored in the string member variable input. One of three options is available. It is as follows:</p> <ol style="list-style-type: none"> <li>(1) Enter a character string, a word, to test the Post machine description.</li> <li>(2) Enter 'N', a null string, to test the Post machine.</li> <li>(3) Enter '*' to turn off the Post machine.</li> </ol> <p>Under case (3), this action sets the Boolean variable off to true. This will cause the machine to turn off. The user is returned to the main menu.</p>
<b>Return Type:</b>	void
<b>Parameters:</b>	void
<b>Pre-condition:</b>	None
<b>Post-condition:</b>	The user supplies a string of one or more characters. If 'N' is entered, the word is a null string. If '*' is entered, the machine is turned off.
<b>Requires:</b>	'N' is required to represent a null string. '*' is required to turn off the Post machine. Otherwise, any string will be tested on the machine.

#### 4.2.1.3.4 start()

<b>Name:</b>	start()
<b>Access:</b>	private
<b>Description:</b>	start() enqueues successive character(s) from the input string into a character queue tape. This is done by successive calls of enqueue(x) , a public member of CharQueue. With the current state i, a look-up function getTransitionTableNextState[i].getNextState() returns an integer index of the table's next state.
<b>Return Type:</b>	start() returns an integer of the next state of the Post machine.
<b>Parameters:</b>	i : Integer is the current state of the Post machine.
<b>Pre-condition:</b>	The current state is start.
<b>Post-condition:</b>	The character queue tape contains a word.
<b>Requires:</b>	The current state is start.

#### 4.2.1.3.5 readTape()

<b>Name:</b>	readTape()
<b>Access:</b>	private
<b>Description:</b>	readTape() dequeues a character from the queue tape. It calls matchAlphabet(Char) to return an integer, an array index associated with the input tape's alphabet. In addition, the index is the position of the transition table's next state, readNextState[j]. If the tape's queue is null, the index is equal to the size of the tape's alphabet. With the current state i and the index j, a look-up table function getTransitionTableReadNextState[i].getReadNext(j) returns an integer of the Post machine's next state.
<b>Return Type:</b>	readtape() returns an integer of the next state of the Post machine.
<b>Parameters:</b>	i : Integer is the current state of the Post machine.
<b>Pre-condition:</b>	The current state is start.
<b>Post-condition:</b>	The next state is found. Otherwise, it is an error.
<b>Requires:</b>	The current state is a read.

#### 4.2.1.3.6 consult()

<b>Name:</b>	consult()
<b>Access:</b>	private
<b>Description:</b>	consult() dequeues a from the queue store. It calls matchAlphabet(Char) to return an integer, an array index associated with the store's alphabet. In addition, the index is the position of the transition table's next state, consultNextState[j]. If the store's queue is null, the index is equal to the size of the store's alphabet. With the current state i and the index j, a look-up table function getTransitionTableConsultNextState[i].getConsultNext(j) returns an integer of the Post machine's next state.
<b>Return Type:</b>	consult() returns an integer of the next state of the Post machine.
<b>Parameters:</b>	i : integer is the current state of the Post machine.
<b>Pre-condition:</b>	The current state is consult.
<b>Post-condition:</b>	The next state is found. Otherwise, it is an error.
<b>Requires:</b>	The current state is a consult.

#### 4.2.1.3.7 storeX()

<b>Name:</b>	storeX()
<b>Access:</b>	private
<b>Description:</b>	storeX() enqueues a character to the rear of the queue store. With the current state i, a look-up table function getTransitionTableNextState[i].getNextState() returns an integer of the Post machine's next state.
<b>Return Type:</b>	storeX() returns an integer of the next state of the Post machine.
<b>Parameters:</b>	i : Integer is the current state of the Post machine.
<b>Pre-condition:</b>	The current state is store.
<b>Post-condition:</b>	The next state is found. Otherwise, it is an error.
<b>Requires:</b>	The current state is a store.

#### 4.2.1.3.8 shiftRightCyclically()

<b>Name:</b>	shiftRightCyclically()
<b>Access:</b>	private
<b>Description:</b>	shiftRightCyclically() shifts each character to the right and the rear character to the front of the queue store. When the rear character is found, it is put onto the front of the queue by addFront() . If the queue is empty, there is no change to the queue. With the current state i, a look-up function getTransitionTableNextState[i].getNextState() returns an integer of the Post machine's next state.
<b>Return Type:</b>	shiftRightCyclically() returns an integer of the next state of the Post machine.
<b>Parameters:</b>	i : Integer is the current state of the Post machine.
<b>Pre-condition:</b>	The current state is shiftRightCyclically.
<b>Post-condition:</b>	The next state is found. Otherwise, it is an error.
<b>Requires:</b>	The current state is a shiftRightCyclically, and the store is not empty.

#### 4.2.1.3.9 addFront()

<b>Name:</b>	addFront()
<b>Access:</b>	private
<b>Description:</b>	addFront() calls the queue operation addFront() to add a character onto the front of the queue store. With the current state i, a look-up table function getTransitionTableNextState[i].getNextState() returns an integer index of the Post machine's next state.
<b>Return Type:</b>	addFront() returns an integer of the next state of the Post machine.
<b>Parameters:</b>	i : Integer is the current state of the Post machine.
<b>Pre-condition:</b>	The current state is add front.
<b>Post-condition:</b>	The next state is found. Otherwise, it is an error.
<b>Requires:</b>	The current state is an add front, and the store is not full.

## 4.2.2 CharQueue class

### 4.2.2.1 Description

CharQueue implements a character queue using a circular array-based structure. The elements are added at the rear of the queue and removed from the front of the queue. There is an integer pointer called front to indicate the front of the queue, and an integer pointer called rear to indicate the rear of the queue. This class does contain some non-conventional operations of the queue:

- (1) printQueue() prints the contents of the character queue.
- (2) src() (shift right cyclically) rotates the queue to the right.
- (3) addFront() adds a character to the front of the queue.

### 4.2.2.2 CharQueue attributes

Name	Object Type	Access	Description
elements[ ]	Character	private	Elements are characters
front	Integer	private	Index of the slot preceding front
rear	Integer	private	Index of slot containing the rear

### 4.2.2.3 CharQueue functions

#### 4.2.2.3.1 makeNullCharQueue()

<b>Name:</b>	makeNullCharQueue()
<b>Access:</b>	public
<b>Description:</b>	makeNullCharQueue() initializes queue to an empty state.
<b>Return Type:</b>	void
<b>Parameters:</b>	void
<b>Pre-condition:</b>	None
<b>Post-condition:</b>	Queue is empty
<b>Requires</b>	None

#### 4.2.2.3.2 emptyCharQueue()

**Name:** emptyCharQueue()  
**Access:** public  
**Description:** emptyCharQueue() indicates whether the queue is empty.  
**Return Type:** Boolean  
**Parameters:** void  
**Pre-condition:** Queue has been initialized.  
**Post-condition:** Empty queue means rear equals front.  
**Requires:** None

#### 4.2.2.3.3 fullCharQueue()

**Name:** fullCharQueue()  
**Access:** public  
**Description:** fullCharQueue() indicates whether the queue is full.  
**Return Type:** Boolean  
**Parameters:** void  
**Pre-condition:** Queue has been initialized.  
**Post-condition:** Full queue means  $\text{increment}(\text{rear} \bmod \text{MAX})$  equals front.  
**Requires:** None

#### 4.2.2.3.4 enqueue()

**Name:** enqueue()  
**Access:** public  
**Description:** enqueue() increments  $(\text{rear} \bmod \text{MAX})$ . Rear may wrap around to the front of the queue.  
**Return Type:** void  
**Parameters:** x : Char is the character to be added to the rear of the queue.  
**Pre-condition:** Queue has been initialized and is not a full queue.  
**Post-condition:** The index rear is first incremented and a character is inserted at the rear of the queue.  
**Requires:** A test for fullCharQueue() needs to be implemented before enqueue() is used.

#### 4.2.2.3.5 dequeue()

<b>Name:</b>	dequeue()
<b>Access:</b>	public
<b>Description:</b>	dequeue() increments (front MOD MAX). Front may wrap around to the rear of the queue. A character is returned from the front of the queue.
<b>Return Type:</b>	dequeue() returns a character from the front of the queue.
<b>Parameters:</b>	void
<b>Pre-condition:</b>	Queue has been initialized and is not an empty queue.
<b>Post-condition:</b>	Front has been incremented front MOD MAX and a character is removed and returned to the calling function.
<b>Requires:</b>	A test for emptyCharQueue() needs to be performed before dequeue() is used.

#### 4.2.2.3.6 printQueue()

<b>Name:</b>	printQueue()
<b>Access:</b>	public
<b>Description:</b>	printQueue() enqueues a special end marker '*' to the rear of the queue. A conditional loop is executed. It dequeues, prints and enqueues each character until the special end marker is encountered.
<b>Return Type:</b>	void
<b>Parameters:</b>	void
<b>Pre-condition:</b>	The queue is not empty.
<b>Post-condition:</b>	The queue contents are not changed.
<b>Requires:</b>	A test for emptyCharQueue() needs to be performed before printQueue() is used.

#### 4.2.2.3.7 addFront()

<b>Name:</b>	addFront()
<b>Access:</b>	public



**Description:** addFront() enqueues a special end marker '\*' to the rear of the queue. It is followed by another enqueue of the x : Char. A conditional loop is executed. It dequeues and enqueues each character until the end marker is encountered. The element that follows the end is the character now on the front of the queue.

**Return Type:** void

**Parameters:** x : Char is the character to be added to the front of the queue.

**Pre-condition:** The queue has been established, and it is not full.

**Post-condition:** x is on the front of the queue

**Requires:** A test for a fullCharQueue() is to be performed before addFront() is used.

#### 4.2.2.3.8 src()

**Name:** src()

**Access:** public

**Description:** src() enqueues a special end marker '\*' to the rear of the queue. src() dequeues a character and stores it in the variable hold. If it is not an end marker, then it dequeues again. If the dequeued element is not an end marker, it enqueues the character in hold, and stores the dequeued character in hold. The process is repeated until it encounters the end marker. Eventually, the character in variable hold is the rear element of the queue. Member function addFront(hold) places the rear character onto the front of the queue.

**Return Type:** void

**Parameters:** void

**Pre-condition:** Queue is established and it is not full.

**Post-condition:** The queue has shifted cyclically to the right, and the rear element is now on the front of the queue.

**Requires:** A test for emptyCharQueue() needs to be performed before src() is to be used.

### 4.2.3 TransitionTable class

#### 4.2.3.1 Description

TransitionTable defines an array structure for the transition table for the Post machine. The table contains a state type, a letter to be added to the store, the next state, an array for the next state if read state is executed, and an array for the next state if consult state is executed. This class is composed of member functions that allow the user to input data into the table and look-up functions that return data from the table.

#### 4.2.3.2 TransitionTable attributes

Name	Object Type	Access	Description
TransitionTable[ ]	TransitionTableField	private	Each row of transition table represents one state of the Post machine
sizeOfTransitionTable	Integer	private	The number of states of the Post machine excluding reject states

#### 4.2.2.3 TransitionTable functions

##### 4.2.2.3.1 inputTransitionTable()

<b>Name:</b>	inputTransitionTable()
<b>Access:</b>	public
<b>Description:</b>	inputTransitionTable() prompts the user to enter an integer the number states of the Post machine excluding its reject states. The user is prompted for each state to select a state type. Once the state is selected. The user is prompted to enter additional information in context with each state type. There are six selections to choose for state type: (1) Read Tape, (2) Consult Queue, (3) Store Queue, (4) Shift Right Cyclically (SRC), (5) Add Front (AF), and (6) Accept.
<b>Return Type:</b>	void
<b>Parameters:</b>	a : Alphabet is the alphabet associated with the input tape (queue), and b : Alphabet is the alphabet associated with the store (queue).

**Pre-condition:** Transition table is initialized.  
**Post-condition:** Each state type is defined, and all the fields of the table are defined.  
**Requires:** User to answer all prompts correctly with appropriate data type.

#### 4.2.2.3.2 printTransitionTable()

**Name:** printTransitionTable()  
**Access:** public  
**Description:** printTransitionTable() prints the contents of the transition table from array index = 0 to index = sizeofTransitionTable + 1. It prints fields for state number, state type, letter, and the next state of the Post Machine.  
**Return Type:** void  
**Parameters:** a : Alphabet is the alphabet associated with the input tape (queue), and b : Alphabet is the alphabet associated with the store (queue).  
**Pre-condition:** Transition table is established and not empty.  
**Post-condition:** None  
**Requires:** None

#### 4.2.2.3.3 getTransitionTableStateType()

**Name:** getTransitionTableStateType()  
**Access:** public  
**Description:** getTransitionTableStateType() is a look-up function that returns a character to indicate the state type of the machine. They are as follows: (1) Reject = X, (2) Start = I, (3) Read = R, (4) Consult = K, (5) Store = W, (6) SRC = S, (7) AF = F, and (8) Accept = A.  
**Return Type:** getTransitionTableStateType() returns a character to indicate the state of the machine.  
**Parameters:** i : Integer is the index of the row of the transition table.  
**Pre-condition:** Transition table is established and  $0 \leq i \leq \text{sizeofTransitionTable} + 1$ .  
**Post-condition:** The returned character must correspond to one of the above state types. Otherwise, it is an error.  
**Requires:** None

#### 4.2.2.3.4 getTransitionTableLetter()

<b>Name:</b>	getTransitionTableLetter()
<b>Access:</b>	public
<b>Description:</b>	getTransitionTableLetter() is a look-up function that returns a letter from the letter field of the transition table at row i.
<b>Return Type:</b>	getTransitionTableLetter() returns a character ( a letter of the store alphabet) to be added to the store (queue).
<b>Parameters:</b>	i : Integer is the index of the row of the transition table.
<b>Pre-condition:</b>	Transition table is established and $0 \leq i \leq \text{sizeOf TransitionTable}+1$ .
<b>Post-condition:</b>	The returned character must correspond to one of the letters of the store alphabet.
<b>Requires:</b>	None

#### 4.2.2.3.5 getTransitionTableNextState()

<b>Name:</b>	getTransitionTableNextState()
<b>Access:</b>	public
<b>Description:</b>	getTransitionTableNextState() is a look-up function that returns an integer from the next state field of the transition table at row i.
<b>Return Type:</b>	getTransitionTableNextState() returns an integer of the next state of the Post machine.
<b>Parameters:</b>	i : Integer is the index of the row of the transition table.
<b>Pre-condition:</b>	Transition table is established and $0 \leq i \leq \text{sizeOf TransitionTable}+1$ .
<b>Post-condition:</b>	The returned integer must be between 0 and $\text{sizeOfTransitionTable}+1$ , inclusive.
<b>Requires:</b>	None

#### 4.2.2.3.6 getTransitionTableReadNextState()

<b>Name:</b>	getTransitionTableReadNextState()
<b>Access:</b>	public
<b>Description:</b>	getTransitionTableReadNextState() is a look-up function that returns an integer from the next state field, readNextState[j], of the transition table at row i.
<b>Return Type:</b>	getTransitionTableReadNextState() returns an integer of the next state of the Post machine.
<b>Parameters:</b>	i : Integer is the index of the row of the transition table, and j : Integer is the index of the column of readNextState[j] (see TransitionTableField class) associated with the input tape's alphabet.
<b>Pre-condition:</b>	Transition table is established and $0 \leq i \leq \text{sizeOfTransitionTable}+1$ .
<b>Post-condition:</b>	An integer is returned that is between 0 and sizeOfTransitionTable +1, inclusive.
<b>Requires:</b>	None

#### 4.2.2.3.7 getTransitionTableConsultNextState()

<b>Name:</b>	getTransitionTableConsultNextState()
<b>Access:</b>	public
<b>Description:</b>	getTransitionTableConsultNextState() is a look-up function that returns an integer from the next state field, consultNextState[j], of the transition table at row i.
<b>Return Type:</b>	getTransitionTableConsultNextState() returns an integer of the next state of the Post machine.
<b>Parameters:</b>	i : Integer is the index of the row of the transition table, and j : Integer is the index of the column of consultNextState[j] (see TransitionTableField class) associated with the store alphabet.
<b>Pre-condition:</b>	Transition table is established and $0 \leq i \leq \text{sizeOfTransitionTable}+1$ .
<b>Post-condition:</b>	The returned integer must be between 0 and sizeOfTransitionTable +1, inclusive.
<b>Requires:</b>	None

#### 4.2.2.3.8 displayLegendForStateType()

<b>Name:</b>	displayLegendForStateType()
<b>Access:</b>	private
<b>Description:</b>	displayLegendForStateType() prints a legend of the transition table. It indicates a state type as follows: (1) Reject = X, (2) Start = I, (3) Accept = A, (4) Read = R, (5) Consult = K, (6) Store = W, (7) Shift Right Cyclically (SRC) = S, and (8) Add Front (AF) = F.
<b>Return Type:</b>	void
<b>Parameters:</b>	void
<b>Pre-condition:</b>	None
<b>Post-condition:</b>	None
<b>Requires:</b>	None

### 4.2.4 TransitionTableField class

#### 4.2.4.1 Description

TransitionTableField class defines the specific fields of the transition table rows. The fields are as follows:

- (1) stateType : Character
- (2) letter : Character
- (3) nextState : Integer
- (4) readNextState[i] : Integer (  $0 \leq i \leq \text{size of input alphabet} + 1$  )
- (5) consultNextState[i] : Integer (  $0 \leq i \leq \text{size of store alphabet} + 1$  )

The class is composed of member functions that allow the user to input data into fields of the table and look-up functions to retrieve data.

#### 4.2.4.2 TransitionTableField attributes

<b>Name</b>	<b>Object Type</b>	<b>Access</b>	<b>Description</b>
stateType	Character	private	Types of states of the Post machine
letter	Character	private	Letter to added to the store queue
nextState	Integer	private	The next state of the transition table
readNextState[ ]	Integer	private	The next state of the transition table indexed with the

ConsultNextState[ ]	Integer	private	input alphabet The next state of the transition table indexed with the store alphabet
---------------------	---------	---------	--

#### 4.2.4.3 TransitionTable functions

##### 4.2.4.3.1 inputStartStateType()

<b>Name:</b>	inputStartStateType()
<b>Access:</b>	public
<b>Description:</b>	inputStartStateType() initializes stateType to I (initial). 'I' is designated as the start state.
<b>Return Type:</b>	void
<b>Parameters:</b>	void
<b>Pre-condition:</b>	The state of the Post machine is a start state.
<b>Post-condition:</b>	The start state equals 'I' and stored in the stateType field.
<b>Requires:</b>	State type 'I' is stored in the second row of the transition table at index 1.

##### 4.2.4.3.2 inputStateType()

<b>Name:</b>	inputStateType()
<b>Access:</b>	public
<b>Description:</b>	inputStateType() displays a menu and prompts the user to choose a state of the Post machine. They are as follows: (1) Read Tape, (2) Consult Queue (Store), (3) Store Queue (Store), (4) Shift Right Cyclically (SRC), (5) Add Front (AF), and (6) Accept. The user enters a number. The menu choice assigns a specific character to a state type. They are as follows: (1) stateType = I (start), (2) stateType = R (read), (3) stateType = K (consult), (4) stateType = W (store), (5) stateType = S (SRC), (6) stateType = F (AF), (7) stateType = X (reject), and (8) stateType = A (accept).
<b>Return Type:</b>	void
<b>Parameters:</b>	void
<b>Pre-condition:</b>	None
<b>Post-condition:</b>	The user enters a state type
<b>Requires:</b>	None

#### 4.2.4.3.3 getStateType()

<b>Name:</b>	getStateType()
<b>Access:</b>	public
<b>Description:</b>	getStateType() is a look-up function that retrieves a character that represents a state of the Post machine
<b>Return Type:</b>	getStateType() returns a character that represents a state type of the Post machine.
<b>Parameters:</b>	void
<b>Pre-condition:</b>	The field is initially established during input operation.
<b>Post-condition:</b>	A character is returned that corresponds to one of the allowed states. Otherwise, it is an error.
<b>Requires:</b>	The field is either initialized to X (reject) or the user must input its state type interactively.

#### 4.2.4.3.4 inputLetter()

<b>Name:</b>	inputLetter()
<b>Access:</b>	public
<b>Description:</b>	inputLetter() prompts the user to enter a letter to add to the store.
<b>Return Type:</b>	void
<b>Parameters:</b>	void
<b>Pre-condition:</b>	The user chooses one of the following, a store queue state or add front state.
<b>Post-condition:</b>	A letter from the store alphabet is stored in the field letter
<b>Requires:</b>	The user is to input a letter of the store alphabet.

#### 4.2.4.3.5 getLetter()

<b>Name:</b>	getLetter()
<b>Access:</b>	public
<b>Description:</b>	getLetter() is a look-up function that retrieves a character that corresponds to one of the letters of the store alphabet
<b>Return Type:</b>	getLetter() returns a character from the transition table field letter that corresponds to one of the letters of the store alphabet.
<b>Parameters:</b>	void



<b>Pre-condition:</b>	The field letter is initially established during input operation. Either the post machine is in a store state or add front state.
<b>Post-condition:</b>	A character is returned that corresponds to one of the letters of the store alphabet. Otherwise, it is an error.
<b>Requires:</b>	Either the post machine is in a store state or add front state.

#### 4.2.4.3.6 inputNextState()

<b>Name:</b>	inputNextState()
<b>Access:</b>	public
<b>Description:</b>	inputNextState() prompts the to enter an integer to branch to the next state.
<b>Return Type:</b>	void
<b>Parameters:</b>	void
<b>Pre-condition:</b>	User selects either store queue, shift right cyclically or add front.
<b>Post-condition:</b>	An integer is stored in the field next State and $0 \leq i \leq \text{sizeofTransitionTable}+1$ . Otherwise, it is an error.
<b>Requires:</b>	User enters an integer between 0 and sizeofTransitionTable, inclusive.

#### 4.2.4.3.7 getNextState()

<b>Name:</b>	getNextState()
<b>Access:</b>	public
<b>Description:</b>	getNextState() is a look-up function that retrieves an integer that corresponds to the next state of the Post machine.
<b>Return Type:</b>	getNextState() returns a character from the transition table field nextState that corresponds to the next state of the Post machine.
<b>Parameters:</b>	void
<b>Pre-condition:</b>	The field nextState is initially established during input operation. The nextState is an integer that is initialized either in start, store, SRC or AF.
<b>Post-condition:</b>	An integer is stored in the field next State and $0 \leq i \leq \text{sizeofTransitionTable}+1$ . Otherwise, it is an error.
<b>Requires:</b>	User enters an integer between 0 and sizeofTransitionTable, inclusive.

#### 4.2.4.3.8 inputReadNextState()

<b>Name:</b>	inputReadNextState()
<b>Access:</b>	public
<b>Description:</b>	InputReadNextState() prompts the to enter an integer to branch to the next state. User is prompted with letters of the input alphabet, and he will answer each alphabet query. For the case of a null query, the user will supply an integer for the next state.
<b>Return Type:</b>	void
<b>Parameters:</b>	a : Alphabet is the alphabet associated with the input tape.
<b>Pre-condition:</b>	User selects the state type read tape.
<b>Post-condition:</b>	An integer is stored in each of the fields of readNextState[i]. The array index i is between 0 and size of the input tape alphabet, inclusive. The integer stored in readNextState[i] is between 0 and sizeOfTransitionTable+1, inclusive.
<b>Requires:</b>	User selects the read state.

#### 4.2.4.3.9 getReadNextState()

<b>Name:</b>	getReadNextState()
<b>Access:</b>	public
<b>Description:</b>	getNextReadState() is a look-up function that retrieves an integer at ReadNextState[i]. An array index i is passed as parameter. This index i corresponds to an associated index of the input tape alphabet. Each letter of the alphabet has a unique integer index between 0 and size of input alphabet, inclusive.
<b>Return Type:</b>	getReadNextState() returns an integer of the next state of the Post machine.
<b>Parameters:</b>	i : Integer is the array index of readNextState[i].
<b>Pre-condition:</b>	Every readNextState[i] is initialized during the corresponding input operation, inputReadNextState().
<b>Post-condition:</b>	The integer returned is between 0 and sizeOfTransitionTable+1, inclusive.
<b>Requires:</b>	The state of the Post machine is in read.

#### 4.2.4.3.10 inputConsultNextState()

<b>Name:</b>	inputConsultNextState()
<b>Access:</b>	public
<b>Description:</b>	inputConsultNextState() prompts the to enter an integer to branch to the next state. User is prompted with letters of the store alphabet, and he will answer each alphabet query. For the case of a null query, the user will supply an integer for the next state.
<b>Return Type:</b>	void
<b>Parameters:</b>	a : Alphabet is the alphabet associated with the store.
<b>Pre-condition:</b>	User selects the state type consult.
<b>Post-condition:</b>	An integer is stored in each of the fields of consultNextState[i]. The array index i is between 0 and size of the store alphabet, inclusive. The integer stored in consultNextState[i] is between 0 and sizeOfTransitionTable+1, inclusive.
<b>Requires:</b>	User selects the consult state.

#### 4.2.4.3.11 getConsultNextState()

<b>Name:</b>	getConsultNextState()
<b>Access:</b>	public
<b>Description:</b>	getConsultReadState() is a look-up function that retrieves an integer at consultNextState[i]. An array index is passed as parameter. This index corresponds to an associated index of the store alphabet. Each letter of the alphabet has a unique integer index between 0 and size of store alphabet, inclusive.
<b>Return Type:</b>	getConsultNextState() returns an integer of the next state of the Post machine.
<b>Parameters:</b>	i : Integer is the array index of consultNextState[i].
<b>Pre-condition:</b>	Every consultNextState[i] is initialized during the corresponding input operation, inputConsultNextState().
<b>Post-condition:</b>	The integer returned is between 0 and sizeOfTransitionTable+1, inclusive.
<b>Requires:</b>	The state of the Post machine is in consult.

#### 4.2.4.3.12 displayStateTypeMenu()

<b>Name:</b>	displayStateTypeMenu()
<b>Access:</b>	private
<b>Description:</b>	displayStateTypeMenu() prompts the user with a menu. The menu choices are as follows: (1) Read Tape, (2) Consult Queue, (3) Store Queue, (4) Shift Right Cyclically (SRC), (5) Add Front (AF), and (6) Accept.
<b>Return Type:</b>	void
<b>Parameters:</b>	void
<b>Pre-condition:</b>	None
<b>Post-condition:</b>	None
<b>Requires:</b>	displayStateTypeMenu() must appear before the state type input.

### 4.2.5 Alphabet class

#### 4.2.5.1 Description

Alphabet is an array-based list of records of a letter of the alphabet and an index.

#### 4.2.5.2 Alphabet attributes

<b>Name</b>	<b>Object Type</b>	<b>Access</b>	<b>Description</b>
Alphabet[]	AlphabetField	private	An array of characters and integers
SizeOfAlphabet	Integer	private	Size of the alphabet

#### 4.2.5.3 Alphabet functions

##### 4.2.5.3.1 makeNullAlphabet()

<b>Name:</b>	makeNullAlphabet()
<b>Access:</b>	public
<b>Description:</b>	makeNullAphabet() sets the alphabet list to an empty state, sizeOfAlphabet = 0.
<b>Return Type:</b>	void
<b>Parameters:</b>	void
<b>Pre-condition:</b>	None
<b>Post-condition:</b>	The alphabet list is initialized to an empty state.
<b>Requires:</b>	None

#### 4.2.5.3.2 matchAlphabet()

<b>Name:</b>	matchAlphabet()
<b>Access:</b>	public
<b>Description:</b>	matchNullAphabet() searches the alphabet list for a letter match and it returns its index.
<b>Return Type:</b>	matchAlphabet() returns an integer index of either the input tape alphabet or the store alphabet.
<b>Parameters:</b>	c : Char is either a letter from the input tape's alphabet or the store's alphabet.
<b>Pre-condition:</b>	c : Char must be an element of either input tape alphabet or the store alphabet.
<b>Post-condition:</b>	The integer returned by matchAlphabet() must be between 0 and sizeofAlphabet, inclusive.
<b>Requires:</b>	The parameter c must be in either the input tape's alphabet or the store's alphabet.

#### 4.2.5.3.3 getAlphabetSize()

<b>Name:</b>	getAlphabetSize()
<b>Access:</b>	public
<b>Description:</b>	getAlphabetSize() returns an integer stored in sizeofAlphabet.
<b>Return Type:</b>	getAlphabetSize() returns an integer of the size of the alphabet.
<b>Parameters:</b>	void
<b>Pre-condition:</b>	Alphabet is established
<b>Post-condition:</b>	The integer returned must be between 0 and sizeofAlphabet, inclusive.
<b>Requires:</b>	None

#### 4.2.5.3.4 getLetter()

<b>Name:</b>	getLetter()
<b>Access:</b>	public
<b>Description:</b>	getLetter() consults the alphabet list and returns a letter that corresponds to array index passed as a parameter.
<b>Return Type:</b>	getLetter() returns a letter (character) at array index i of alphabet[i].
<b>Parameters:</b>	i : Integer is the index of the alphabet list
<b>Pre-condition:</b>	The array index must be between 0 and sizeoftheAlphabet, inclusive.
<b>Post-condition:</b>	The returned letter must be a member of the alphabet.

**Requires:** None

#### 4.2.5.3.5 inputAlphabet()

**Name:** inputAlphabet()  
**Access:** public  
**Description:** inputAlphabet() prompts user to input size of alphabet and enter its characters (letters).  
**Return Type:** void  
**Parameters:** void  
**Pre-condition:** None  
**Post-condition:** The size of the alphabet is greater than zero and less the maximum size of the array allocation MAX.  
**Requires:** User cannot enter the special character '\*' or '\$'.

#### 4.2.5.3.6 printAlphabet()

**Name:** printAlphabet()  
**Access:** public  
**Description:** printAlphabet() prints the list of characters of its alphabet.  
**Return Type:** void  
**Parameters:** void  
**Pre-condition:** The alphabet exists.  
**Post-condition:** None  
**Requires:** None

### 4.2.6 AlphabetField class

#### 4.2.6.1 Description

AlphabetField class consists of two fields, letter and index, of the alphabet list (see Alphabet class). The class has two types of operations of input functions and look-up functions.

#### 4.2.6.2 AlphabetField attributes

Name	Object Type	Access	Description
letter	Character	private	Letter of the alphabet
index	Integer	private	An array index of the alphabet

### 4.2.6.3 AlphabetField functions

#### 4.2.6.3.1 inputAlphabetField()

<b>Name:</b>	inputAlphabetField()
<b>Access:</b>	public
<b>Description:</b>	inputAlphabetField() prompts the user to enter a letter of the alphabet. The index parameter notes its position in the array.
<b>Return Type:</b>	void
<b>Parameters:</b>	i : Integer is the array index of alphabet[i].
<b>Pre-condition:</b>	The size of the alphabet is greater than or equal to zero and the index i be between 0 and sizeOfAlphabet, inclusive.
<b>Post-condition:</b>	The fields contain a letter of the input tape's alphabet or store's alphabet. Each letter has a unique index between 0 and sizeOfAlphabet, inclusive.
<b>Requires:</b>	The user must specify the letter.

#### 4.2.6.3.2 getLetter()

<b>Name:</b>	getLetter()
<b>Access:</b>	public
<b>Description:</b>	getLetter() is a look-up function that returns a letter of the alphabet.
<b>Return Type:</b>	getLetter() returns a character of the alphabet
<b>Parameters:</b>	void
<b>Pre-condition:</b>	The alphabet is established.
<b>Post-condition:</b>	The letter belongs to the user-defined alphabet.
<b>Requires:</b>	None

#### 4.2.5.3.3 getIndex()

<b>Name:</b>	getIndex()
<b>Access:</b>	public
<b>Description:</b>	getIndex() returns the array index of the alphabet.
<b>Return Type:</b>	getIndex() returns the array index of the alphabet.
<b>Parameters:</b>	void
<b>Pre-condition:</b>	Alphabet is established
<b>Post-condition:</b>	The integer returned is bounded between 0 and size of the alphabet, inclusive.
<b>Requires:</b>	None





## Post Machine: CHAPTER 5



Ronald Eggers [ronald.eggerts@gte.net](mailto:ronald.eggerts@gte.net)  
Qian Chang [qchang@dhvx20.csudh.edu](mailto:qchang@dhvx20.csudh.edu)  
Mando Gomez [mandolux@yahoo.com](mailto:mandolux@yahoo.com)

---

### Frequently Asked Questions

- Will PostMachine v1.1 be able to use any word for a WORD input?

Yes. As long you define the ALPHABET and the size of it, you should not encounter any problems. You can even use binary digits if you pleased.

- 2) Will PostMachine v1.1 be able to simulate the behavior of a Finite State machine?

PostMachine v1.1 is a Finite State Machine.

- Will PostMachine v1.1 be able to simulate a Turing Machine?

PostMachine v1.1 is a Turing Machine.

- What is a Turing Machine?

A Turing Machine is a machine that can simulate other machines.

- What the hell is a post machine?

A Post machine is an abstract machine consisting of a set of states (including the initial state), a set of input events, a set of output events, and a state transition function.

What is an abstract machine?

- 5) A processor design which is not intended to be implemented as hardware, but which is the notional executor of a particular intermediate language (abstract machine language) used in a compiler or interpreter. An abstract machine has an instruction set, a register set and a model of memory. It may provide instructions which are closer to the language being compiled than any physical computer or it may be used to make the language implementation easier to port to other platforms.
- 6) A procedure for executing a set of instructions in some formal language, possibly also taking in input data and producing output. Such abstract machines are not intended to be constructed as hardware but are used in thought experiments about computability.

For more information, please see CHAPTER 1.

- Does this have anything to do with breakfast cereal?

Nope.

- Will this replace the guy who brings me snail mail every day?

We wish... but it doesn't.

- What is a context free grammar, and why didn't they teach me that in English class?

Context-Free Grammars (CFG) is a collection of three things:

- 1) An alphabet  $\Sigma$  of letters called terminals from which we are going to make strings that will be the words of a language.
- 11) A set of symbols called non-terminals, one of which is the symbol  $S$ , standing for 'Start here.'
- 12) A finite set of productions of the form:  
One Non-terminals  $\rightarrow$  finite strings of terminals and/or Non-terminals

Where the strings of terminal, and non-terminal can consists of only terminals or of only non-terminals, or of any mixture of terminals and non-terminals or even the empty string. We require that at least one production has the non-terminal  $S$  as its left side.

- The Post machine takes longer to run than a finite state acceptor or deterministic push down acceptor, so how can it be better?

It is better is the way it stores the data and how it deals with it.

- Why does having a Store (ADT queue) make the Post machine more powerful than a deterministic push down acceptor?

Because it does! If you are asking this question, it means you didn't read one thing before this chapter. Shame on you!

- Will knowing how to program a Post machine help me pay my bills?

Well, if you play with how a Post machine processes its input, you can convert it so it can tell you what stocks to buy!

# Post Machine: APPENDIX



Ronald Eggers [ronald.eggert@gte.net](mailto:ronald.eggert@gte.net)  
Qian Chang [qchang@dhvx20.csudh.edu](mailto:qchang@dhvx20.csudh.edu)  
Mando Gomez [mandolux@yahoo.com](mailto:mandolux@yahoo.com)

## APPENDIX

### PostMachine v1.1 Source Code

```
// Mathematics 361 Finite Automata
// Term Project: Simulation of Post Machine
//
// Group Participants:
//
// Qian Chang
// Ron Eggers
// Mando Gomez
//
// Due Date: May, 2000
//
// For detailed documentation consult analyst's notes and user's manual
//
// Note: There is error no exception handling as far entering the correct
//       data types. For this program to execute and be useful. The user
//       must enter the correct types when prompted. There are two types
//       of data required interactively: (1) integer input to indicate a state
//       number, size of alphabet, or size of transition table and
//       (2) character input to indicate the alphabet of the input tape or store.
//       This must be done as requested otherwise this program will crash.
//
//
#include <iostream.h>
#include <stdlib.h>
#include <string.h>
#include <iomanip.h>
#define MAXQUEUE 250 // MAXQUEUE is the upper bound on the
//                    // maximum number of
//                    // plus one
characters in the

#define MAX 10

class CharQueue{
private:
    char elements[MAXQUEUE]; // CharQueue of characters
    int front; // array index of slot preceding
front
    int rear; // index of slot containing rear

public:

    // Constructor initailizes CharQueue to empty state

    CharQueue()
    {
        front = 0;
        rear = 0;
    }

    // makenullCharQueue() initailizes CharQueue to empty state
```

```

void makenullCharQueue()
{
    front = 0;
    rear = 0;
}

// emptyCharQueue() indicates whether queue is empty

bool emptyCharQueue()
{
    return front == rear;
}

// fullCharQueue() indicates whether queue is full

bool fullCharQueue()
{
    return ((rear % MAXQUEUE) + 1) == front;
}

// enqueue(char x) adds a new character to rear of queue

void enqueue(char x)
{
    rear = (rear % MAXQUEUE) + 1;
    elements[rear] = x;
}

// dequeue() removes the front element from the queue and returns it

char dequeue()
{
    front = (front % MAXQUEUE) + 1;
    return elements[front];
}

// printQueue() prints the contents of the queue

void printQueue()
{
    char c;
    if (!fullCharQueue())
        enqueue('@');
    while ( (c = dequeue()) != '@' )
    {
        cout << c;
        enqueue(c);
    }
}

// addFront() adds a charcter to the front of the queue

void addFront(char x)
{
    char c;
    if (!fullCharQueue())
        enqueue('@'); // enqueue an endmarker '@'
    else
    {
        cout << "\nFull Queue\n";
        exit(1);
    }
    if (!fullCharQueue())
        enqueue(x); // enqueue a character for
the front
    else
    {
        cout << "\nFull Queue\n";
        exit(1);
    }
}

```

```

        c = dequeue();
        while( c != '@')           // stops when character is at front
        {
            enqueue(c);
            c=dequeue();
        }

    } // add Front

    // src() rotates the queue to the right. Rear is now front.

void src()
{
    char hold, c;
    bool finished=false;
    if(!fullCharQueue())
        enqueue('@');
    else
    {
        cout << "\nFull Queue\n";
        exit(1);
    }
    hold=dequeue();           // hold character if not the last in queue
    if (hold != '@') // Not an empty queue
    {
        while(!finished)
        {
            c=dequeue();
            if ( c != '@')
            {
                enqueue(hold);
                hold=c;
            }
            else
                finished=true;
        } // while
        addFront(hold);
    }
} // src()

}; // class CharQueue

class AlphabetField{
private:
    char letter;           // letter of alphabet
    int index;             // array index of letter

public:

    // Constructor

    AlphabetField()
    {
        letter = ' ';
        index = 0;
    }

    // inputAlphabetField() prompts user for an alphabet

    void inputAlphabetField(int i)
    {
        cout << "Enter letter: ";
        cin >> letter;
        index = i;
    }

    char getLetter()

```

```

        {
            return letter;
        }

        int getIndex()
        {
            return index;
        }

}; // class AlphabetField

class Alphabet{
private:
    AlphabetField alphabet[MAX];
    int sizeofAlphabet;

public:
    Alphabet()
    {
        sizeofAlphabet=0;
    }

    void makeNullAlphabet()
    {
        sizeofAlphabet=0;
    }

    int matchAlphabet(char c)
    {
        for( int i=0; i < sizeofAlphabet; i++)
            if( alphabet[i].getLetter() == c )
                return alphabet[i].getIndex();

        return -1;
    }

    int getAlphabetSize()
    {
        return sizeofAlphabet;
    }

    char getLetter(int i)
    {
        return alphabet[i].getLetter();
    }

    void inputAlphabet()
    {
        bool inputOkay = false;
        while ( !inputOkay )
        {
            cout << "\nEnter the size of the alphabet: ";
            cin >> sizeofAlphabet;
            if (((sizeofAlphabet >=0) && (sizeofAlphabet <= MAX) ))
                inputOkay = true;
            else
                cout << "\nError: Alphabet size is out of bounds.\n";
        }

        for( int i = 0; i<sizeofAlphabet; i++ )
            alphabet[i].inputAlphabetField(i);
    }

    void printAlphabet()
    {
        cout << "\nAlphabet:\n";
        for (int i = 0; i < sizeofAlphabet; i++)
            cout << "Letter " << i+1 << ": " << alphabet[i].getLetter() <<
endl;

```

```

    }

}; // class Alphabet

class TransitionTableField
{
private:
    char stateType;
    char letter;
    int nextState;
    int readNextState[MAX];
    int consultNextState[MAX];

public:
    // public
    TransitionTableField()
    {
        stateType = 'X'; // X stands for REJECT state
        letter = '-';
        nextState = 0;

        for ( int i = 0; i < MAX; i++ )
            readNextState[i]=0;

        for ( i = 0; i < MAX; i++ )
            consultNextState[i]=0;
    }

    void initializeTransitionTableField()
    {
        stateType = 'X'; // X stands for REJECT state
        letter = '-';
        nextState = 0;

        for ( int i = 0; i < MAX; i++ )
            readNextState[i]=0;

        for ( i = 0; i < MAX; i++ )
            consultNextState[i]=0;
    }

    //public
    void inputStartStateType()
    {
        stateType = 'I'; // Start State
    }

    // public
    void inputStateType()
    {
        int menuNumber;
        bool inputOkay = false;
        while( !inputOkay )
        {
            displayStateTypeMenu();
            cin >> menuNumber;
            if ( ! ( inputOkay = isInBounds( menuNumber, 1, 6 ) ) )
                cout << "\nError: number entered is not a choice.\n";
        }

        switch(menuNumber)
        {
            case 1: stateType = 'R'; // Read input tape (queue)
                     break;
            case 2: stateType = 'K'; // Consult front of store queue
                     break;
            case 3: stateType = 'W'; // Store to end of store queue
                     break;
        }
    }
};

```

```

        case 4:      stateType = 'S'; // Shift queue to the right
                     break;
        case 5:      stateType = 'F'; // add to front of queue
                     break;
        case 6:      stateType = 'A'; // accept state
                     break;
        default:     stateType = '-'; // Not defined
                     break;
    }

}

char getStateType()
{
    return stateType;
}

void inputLetter(Alphabet a)
{
    bool inputOkay = false;
    while ( !inputOkay )
    {
        cout << "Enter letter to add at end of queue: ";
        cin >> letter;
        if ( !(inputOkay==isInAlphabet(letter, a)) )
            cout << "Error: Letter not in Alphabet.\n";
    }
}

char getLetter()
{
    return letter;
}

void inputNextState(int tableSize)
{
    bool inputOkay = false;
    while ( !inputOkay )
    {
        cout << "\nEnter the next state (integer) : ";
        cin >> nextState;
        if ( ! (inputOkay = isInBounds(nextState, 0, tableSize) ) )
            cout << "\nError: State out of bounds of table.\n";
    }
}

int getNextState()
{
    return nextState;
}

void inputReadNextStates(Alphabet a, int tableSize)
{
    bool inputOkay;
    cout << "\nEnter Next State (integer) :\n";
    for( int i = 0; i < a.getAlphabetSize(); i++)
    {
        inputOkay = false;
        while ( !inputOkay )
        {
            cout << "    If " << a.getLetter(i) << ": ";
            cin >> readNextState[i];
            if ( ! ( inputOkay = isInBounds( readNextState[i], 0,
tableSize ) ) )
                cout << "\nError: State out of bounds of table.\n";
        }
    }

    inputOkay = false;
}

```



```

        while ( !inputOkay )
        {
            cout << "    If NULL: ";
            cin >> readNextState[a.getAlphabetSize()];
            if ( ! ( inputOkay = isInBounds(
readNextState[a.getAlphabetSize()], 0, tableSize ) ) )
                cout << "\nError: State out of bounds of table.\n";
        }
    }

    int getReadNextState(int i)
    {
        return readNextState[i];
    }

    void inputConsultNextStates(Alphabet a, int tableSize)
    {
        bool inputOkay;
        cout << "\nEnter Next State (integer) :\n";
        for( int i = 0; i < a.getAlphabetSize(); i++)
        {
            inputOkay = false;
            while ( !inputOkay )
            {
                cout << "        If " << a.getLetter(i) << ": ";
                cin >> consultNextState[i];
                if ( ! ( inputOkay = isInBounds( consultNextState[i], 0,
tableSize ) ) )
                    cout << "\nError: State out of bounds of table.\n";
            }
        }

        inputOkay = false;
        while ( !inputOkay )
        {
            cout << "    If NULL: ";
            cin >> consultNextState[a.getAlphabetSize()];
            if ( ! ( inputOkay = isInBounds(
consultNextState[a.getAlphabetSize()], 0, tableSize ) ) )
                cout << "\nError: State out of bounds of table.\n";
        }
    }

    int getConsultNextState(int i)
    {
        return consultNextState[i];
    }

private:
    void displayStateTypeMenu()
    {
        cout << "\nState Type:\n"
            << "1: Read Tape\n"
            << "2: Consult Queue\n"
            << "3: Store Queue\n"
            << "4: Shift Right Cyclically (SRC)\n"
            << "5: Add Front (AF)\n"
            << "6: Accept State\n\n"
            << "Enter a number: ";
    }

    bool isInBounds( int x, int y, int z )
    {
        if ( x >= y && x <= z )
            return true;
        else

```

```

        return false;
    }

    bool isInAlphabet( char x, Alphabet a)
    {
        for ( int i=0; i<a.getAlphabetSize(); i++)
            if ( x == a.getLetter(i) )
                return true;
        return false;
    }
}; // TransitionTableField

class TransitionTable
{
private:
    TransitionTableField transitionTable[100];
    int sizeOfTransitionTable;
public:
    TransitionTable()
    {
        sizeOfTransitionTable = 0;
    }

    int getSizeOfTransitionTable()
    {
        return sizeOfTransitionTable;
    }

    void makeNullTransitionTable()
    {
        sizeOfTransitionTable = 0;
        for ( int i = 0; i < 100; i++ )
            transitionTable[i].initializeTransitionTableField();
    }

    void inputTransitionTable(Alphabet a, Alphabet b)
    {
        bool inputOkay = false;

        cout << "\nDefining Post Machine:\n\n";
        cout << "\nAll Reject States are defined as 0 (zero).\n\n";
        while ( !inputOkay )
        {
            cout << "Enter the number of states excluding rejected states: ";
            cin >> sizeOfTransitionTable;
            if ( !( inputOkay = isInBounds(sizeOfTransitionTable,0, 99)) )
                cout << "\nError: size entered is out of bounds.\n";
        }
        cout << "\nState 1: START\n";
        transitionTable[1].inputStartStateType();
        transitionTable[1].inputNextState(sizeOfTransitionTable);

        for ( int i =2; i< sizeOfTransitionTable+1; i++)
        {
            cout << "\nState " << i << ":\n";
            transitionTable[i].inputStateType();
            switch(transitionTable[i].getStateType() )
            {
            case 'A': // Accept State
                break;
            case 'R': // Read State
                transitionTable[i].inputReadNextStates(a,
sizeOfTransitionTable);
                break;
            case 'K': // Consult Queue
                transitionTable[i].inputConsultNextStates(b,
sizeOfTransitionTable);
                break;
            case 'W': // Store Queue

```

```

        transitionTable[i].inputLetter(b);
        transitionTable[i].inputNextState(sizeofTransitionTable);
        break;
    case 'S': // Shift Right Cyclically
        transitionTable[i].inputNextState(sizeofTransitionTable);
        break;
    case 'F': // Add Front of Queue
        transitionTable[i].inputLetter(b);
        transitionTable[i].inputNextState(sizeofTransitionTable);
        break;
    default:
        break;
}

    } // for i
} // inputTransitionTable

void printTransitionTable(Alphabet a, Alphabet b)
{
    cout << "\nPrint Transition Table:\n\n";
    cout << "Number State Add Queue Next ";

    for (int j = 0; j < a.getAlphabetSize(); j++)
        cout << " If" << setw(2) << a.getLetter(j);
    cout << " If NULL ";

    if ( b.getAlphabetSize() == 0 )
        cout << endl;

    for ( j = 0; j < b.getAlphabetSize(); j++)
        cout << " If" << setw(2) << b.getLetter(j);

    if ( b.getAlphabetSize() != 0 )
        cout << " If NULL\n";

    for (int i = 0; i < sizeofTransitionTable+1; i++ )
    {
        cout << setw(3) << i
            << setw(8) << transitionTable[i].getStateType()
            << setw(8) << transitionTable[i].getLetter()
            << setw(8) << transitionTable[i].getNextState();

        for( j=0; j < a.getAlphabetSize()+1; j++)
            cout << setw(6) << transitionTable[i].getReadNextState(j);

        if ( b.getAlphabetSize() != 0 )
            for( j=0; j < b.getAlphabetSize()+1; j++)
                cout << setw(6) << transitionTable[i].getConsultNextState(j);

        cout << endl;
    }

    displayLegendForStateType();
}

char getTransitionTableStateType(int i)
{
    return transitionTable[i].getStateType();
}

char getTransitionTableLetter(int i)
{
    return transitionTable[i].getLetter();
}

int getTransitionTableNextState(int i)

```

```

        {
            return transitionTable[i].getNextState();
        }

int getTransitionTableReadNextState(int i, int j)
{
    return transitionTable[i].getReadNextState(j);
}

int getTransitionTableConsultNextState(int i, int j)
{
    return transitionTable[i].getConsultNextState(j);
}

private:

void displayLegendForStateType()
{
    cout << "\n\nLegend for State Type:\n\n"
        << "I = Start,    A = Accept,    X=Reject\n"
        << "R = Read,      K = Consult,    W = Store,\n"
        << "S=Shift Right Cyclically,    F = Add Front\n";
}

bool isInBounds( int x, int y, int z )
{
    if ( x >= y && x <= z )
        return true;
    else
        return false;
}

}; // class TransitionTable

class PostMachine
{
private:
    Alphabet a;
    Alphabet b;
    TransitionTable t;
    CharQueue store;
    CharQueue tape;
    char input[120];
    bool off;

public:

    PostMachine()
    {
        off = false;
    }

    void resetPostMachine()
    {
        off = false;
    }

    void runPM()
    {
        //char c;
        bool finalState=false;
        int iter;

        a.makeNullAlphabet();
        b.makeNullAlphabet();
        t.makeNullTransitionTable();

        cout << "\nEnter Tape Alphabet:\n";
        a.inputAlphabet();
        //a.printAlphabet();
    }

```

```

        cout << "\nEnter Store Alphabet:\n";
        cout << "If all states are read only, then Store alphabet is not
applicable, Enter 0.\n\n";

        b.inputAlphabet();

        //if ( b.getAlphabetSize() != 0 )
        //    b.printAlphabet();

        t.inputTransitionTable(a,b);
        t.printTransitionTable(a,b);

        int i;

        inputWord();

        while ( !off )
        {
            store.makenullCharQueue();
            tape.makenullCharQueue();

            finalState=false;
            i=1;
            iter = 0;
            while ( !finalState && iter < 1000 )
            {
                switch(t.getTransitionTableStateType(i))
                {
                    case 'I': // Start State

                        i=start(i);
                        break;

                    case 'R': // Read tape (queue)
                        i=readTape(i);
                        break;

                    case 'K': // Read or consult State (dequeue)

                        i=consult(i);
                        break;

                    case 'W': // Add state or write state (enqueue)

                        i=storeX(i);
                        break;

                    case 'S': // read back of queue or shift right
cyclically

                        i=shiftRightCyclically(i);
                        break;

                    case 'F': // Add Front State

                        i=addFront(i);
                        break;

                    case 'A': // Accept State

                        cout << "State = " << setw(2) << i << "
ACCEPT\n";

                        finalState=true;
                        break;
                    case 'X': // Reject State
                        cout << "State = " << setw(2) << i << "
REJECT\n";

                        finalState=true;
                        break;
                }
            }
        }

```

```

        default:
            break;
    } // switch

    iter++;

    if (iter%20 == 0)
    {
        cout << "\nPausing output. Hit any key. --> ";
        cin.get(); cin.get();
    }

    } // while ( !finalState )

    inputWord();

    } // while ( !off )

} // class PostMachine

private:

void inputWord()
{
    cout << "\nEnter word or 'N' for null string or '*' to turn off machine:
";
    cin >> input;
    if ( input[0] == '*' )
        off = true;
}

int start(int i)
{
    if( input[0] != 'N' )
    {
        for ( int k=0; k < (int)strlen(input); k++)
            if( !tape.fullCharQueue())
                tape.enqueue(input[k]);
        else
        {
            cout << "\nFull Queue\n";
            exit(1);
        }
        cout << "\nState = " << setw(2) << i << " START      Tape = ";
        tape.printQueue(); cout << "      Store = NULL\n";
    }
    else
        cout << "\nState = " << setw(2) << i << " START      Tape = NULL
Store = NULL\n";
    return t.getTransitionTableNextState(i);
}

int readTape(int i)
{
    char c;
    int j;
    if( !tape.emptyCharQueue())
    {
        c=tape.dequeue();
        j = a.matchAlphabet(c);
    }
    else
        j=a.getAlphabetSize();

    if( !tape.emptyCharQueue())
    {
        cout << "State = " << setw(2) << i << " READ      Tape = ";
        tape.printQueue();
    }
    else

```

```

        cout << "State = " << setw(2) << i << " READ          Tape = NULL ";

    if ( !store.emptyCharQueue() )
    {
        cout << "      Store = ";
        store.printQueue();
        cout << endl;
    }
    else
        cout << "      Store = NULL\n";

    return t.getTransitionTableReadNextState(i,j);
} // int readTape(int i)

int consult(int i)
{
    char c;
    int j;
    if ( !store.emptyCharQueue() )
    {
        c=store.dequeue();
        j=b.matchAlphabet(c);
    }
    else
        j=b.getAlphabetSize();

    if( !tape.emptyCharQueue() )
    {
        cout << "State = " << setw(2) << i << " CONSULT      Tape = ";
        tape.printQueue();
    }
    else
        cout << "State = " << setw(2) << i << " CONSULT      Tape = NULL ";

    if ( !store.emptyCharQueue() )
    {
        cout << "      Store = ";
        store.printQueue();
        cout << endl;
    }
    else
        cout << "      Store = NULL\n";

    return t.getTransitionTableConsultNextState(i,j);
}

int storeX(int i)
{
    if ( !store.fullCharQueue() )
    {
        store.enqueue(t.getTransitionTableLetter(i));
        if ( !tape.emptyCharQueue() )
        {
            cout << "State = " << setw(2) << i << " STORE          Tape =
";

            tape.printQueue();
            cout << "      Store = ";
            store.printQueue();
            cout << endl;
        }
        else
        {
            cout << "State = " << setw(2) << i << " STORE          Tape =
NULL ";

            cout << "      Store = ";
            store.printQueue();
            cout << endl;
        }
    }

    return t.getTransitionTableNextState(i);
}

```

```

    }
    else
    {
        cout << "\nFull Queue\n";
        exit(1);
    }
}

int shiftRightCyclically(int i)
{
    if ( !store.emptyCharQueue() )
        store.src();

    if( !tape.emptyCharQueue() )
    {
        cout << "State = " << setw(2) << i << " SRC          Tape = ";
        tape.printQueue();
    }
    else
        cout << "State = " << setw(2) << i << " SRC          Tape = NULL ";

    if ( !store.emptyCharQueue() )
    {
        cout << "      Store = ";
        store.printQueue();
        cout << endl;
    }
    else
        cout << "      Store = NULL\n";

    return t.getTransitionTableNextState(i);
}

int addFront(int i)
{
    store.addFront(t.getTransitionTableLetter(i));
    cout << "State = " << setw(2) << i << " AF          Store = ";
    store.printQueue(); cout << endl;
    return t.getTransitionTableNextState(i);
}

}; // class PostMachine

void main()
{
    PostMachine pm;
    int i=1;
    while ( i != 2 )
    {
        cout << "\nMenu:\n\n"
              << "1: Run Post Machine\n"
              << "2: Quit\n\n"
              << "Enter: ";

        cin >> i;
        switch(i)
        {
            case 1:
                pm.resetPostMachine();
                pm.runPM();
                break;

            default:
                i=2;
                break;
        }
    }

    cout << "Test to see if this program executes\n";
    cin.get();
}

```

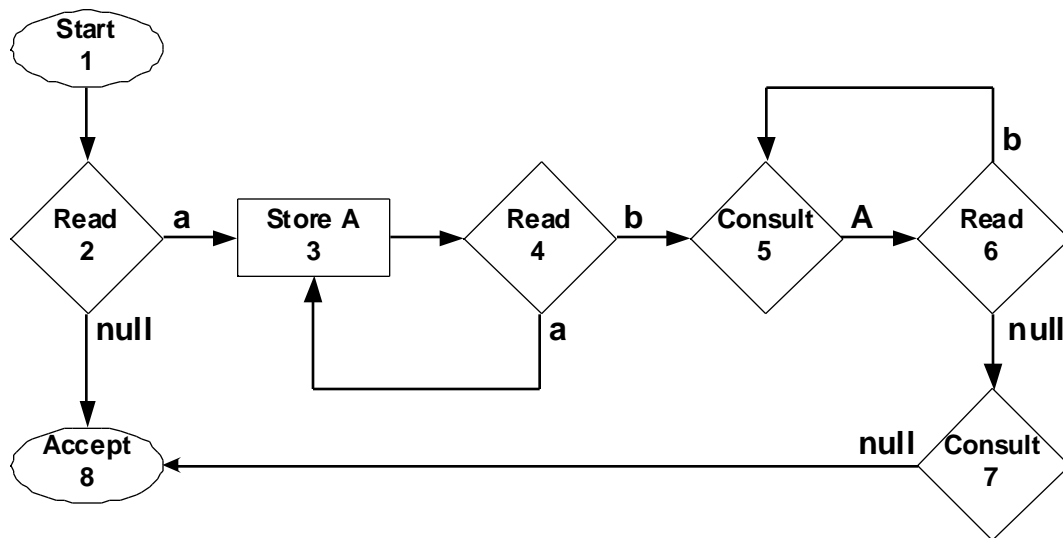


**More sample runs of PostMachine v1.1 ... masochist welcome ...**  
-----**a<sup>n</sup>b<sup>n</sup>**-----  
-----**a<sup>n</sup>b<sup>n</sup>\_01**-----  
-----**a<sup>n</sup>b<sup>n</sup>c<sup>n</sup>**-----  
-----**notaaa**-----  
-----**Palindromes**-----  
-----**Palindromes Cont.**-----

$$a^n b^n$$

**Table 1. Post Machine describing the language  $\{ a^n b^n \mid n \geq 0 \}$**

Number	State Type	Add Letter	Next State	Read/Branch			Consult/Branch	
				If a	If b	If null	If A	If null
0	Reject							
1	Start		2					
2	Read			3	0	8		
3	Store	A	4					
4	Read			3	5	0		
5	Consult						6	0
6	Read			0	5	7		
7	Consult						0	8
8	Accept							



Menu:

1: Run Post Machine  
2: Quit

Enter: 1

Enter Tape alphabet:

Enter the size of the alphabet: 2  
Enter letter: a  
Enter letter: b

Alphabet:  
Letter 1: a  
Letter 2: b

Enter Store alphabet:

Enter the size of the alphabet: 2  
Enter letter: A  
Enter letter: B

Alphabet:  
Letter 1: A  
Letter 2: B

Defining Post Machine:

All Reject States are defined as 0 (zero).

Enter the number of states excluding rejected states: 8

State 1: START

Enter the next state (integer) : 2

State 2:

State Type:  
1: Read Tape  
2: Consult Queue  
3: Store Queue  
4: Shift Right Cyclically (SRC)  
5: Add Front (AF)  
6: Accept State

Enter a number: 1

Enter Next State (integer) :  
If a: 3  
If b: 0  
If NULL: 8

State 3:

State Type:  
1: Read Tape  
2: Consult Queue  
3: Store Queue  
4: Shift Right Cyclically (SRC)  
5: Add Front (AF)  
6: Accept State

Enter a number: 3  
Enter letter to add at end of queue: A

Enter the next state (integer) : 4

State 4:

State Type:  
1: Read Tape  
2: Consult Queue  
3: Store Queue  
4: Shift Right Cyclically (SRC)  
5: Add Front (AF)

```

6: Accept State

Enter a number: 1

Enter Next State (integer) :
    If a: 3
    If b: 5
    If NULL: 0

State 5:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 2

Enter Next State (integer) :
    If A: 6
    If B: 0
    If NULL: 0

State 6:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 1

Enter Next State (integer) :
    If a: 0
    If b: 5
    If NULL: 7

State 7:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 2

Enter Next State (integer) :
    If A: 0
    If B: 0
    If NULL: 8

State 8:

State Type:
1: Read Tape
2: Consult Queue

```

3: Store Queue  
 4: Shift Right Cyclically (SRC)  
 5: Add Front (AF)  
 6: Accept State

Enter a number: 6

Print Transition Table:

Number	State	Add	Queue	Next	If a	If b	If NULL	If A	If B	If NULL
0	X	-		0	0	0	0	0	0	0
1	I	-		2	0	0	0	0	0	0
2	R	-		0	3	0	8	0	0	0
3	W	A		4	0	0	0	0	0	0
4	R	-		0	3	5	0	0	0	0
5	K	-		0	0	0	0	6	0	0
6	R	-		0	0	5	7	0	0	0
7	K	-		0	0	0	0	0	0	8
8	A	-		0	0	0	0	0	0	0

Legend for State Type:

I = Start,      A = Accept,      X=Reject  
 R = Read,      K = Consult,      W = Store,  
 S=Shift Right Cyclically,      F = Add Front

Enter word or 'N' for null string or '\*' to turn off machine: N

State = 1 START      Tape = NULL      Store = NULL  
 State = 2 READ      Tape = NULL      Store = NULL  
 State = 8 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: a

State = 1 START      Tape = a      Store = NULL  
 State = 2 READ      Tape = NULL      Store = NULL  
 State = 3 STORE      Tape = NULL      Store = A  
 State = 4 READ      Tape = NULL      Store = A  
 State = 0 REJECT

Enter word or 'N' for null string or '\*' to turn off machine: b

State = 1 START      Tape = b      Store = NULL  
 State = 2 READ      Tape = NULL      Store = NULL  
 State = 0 REJECT

Enter word or 'N' for null string or '\*' to turn off machine: ab

State = 1 START      Tape = ab      Store = NULL  
 State = 2 READ      Tape = b      Store = NULL  
 State = 3 STORE      Tape = b      Store = A  
 State = 4 READ      Tape = NULL      Store = A  
 State = 5 CONSULT      Tape = NULL      Store = NULL  
 State = 6 READ      Tape = NULL      Store = NULL  
 State = 7 CONSULT      Tape = NULL      Store = NULL  
 State = 8 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: aab

State = 1 START      Tape = aab      Store = NULL  
 State = 2 READ      Tape = ab      Store = NULL  
 State = 3 STORE      Tape = ab      Store = A  
 State = 4 READ      Tape = b      Store = A  
 State = 3 STORE      Tape = b      Store = AA

State =	4 READ	Tape = NULL	Store = AA
State =	5 CONSULT	Tape = NULL	Store = A
State =	6 READ	Tape = NULL	Store = A
State =	7 CONSULT	Tape = NULL	Store = NULL
State =	0 REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: aabbb

State =	1 START	Tape = aabbb	Store = NULL
State =	2 READ	Tape = abbb	Store = NULL
State =	3 STORE	Tape = abbb	Store = A
State =	4 READ	Tape = bbb	Store = A
State =	3 STORE	Tape = bbb	Store = AA
State =	4 READ	Tape = bb	Store = AA
State =	5 CONSULT	Tape = bb	Store = A
State =	6 READ	Tape = b	Store = A
State =	5 CONSULT	Tape = b	Store = NULL
State =	6 READ	Tape = NULL	Store = NULL
State =	5 CONSULT	Tape = NULL	Store = NULL
State =	0 REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: aaabbb

State =	1 START	Tape = aaabbb	Store = NULL
State =	2 READ	Tape = aabbb	Store = NULL
State =	3 STORE	Tape = aabbb	Store = A
State =	4 READ	Tape = abbb	Store = A
State =	3 STORE	Tape = abbb	Store = AA
State =	4 READ	Tape = bbb	Store = AA
State =	3 STORE	Tape = bbb	Store = AAA
State =	4 READ	Tape = bb	Store = AAA
State =	5 CONSULT	Tape = bb	Store = AA
State =	6 READ	Tape = b	Store = AA
State =	5 CONSULT	Tape = b	Store = A
State =	6 READ	Tape = NULL	Store = A
State =	5 CONSULT	Tape = NULL	Store = NULL
State =	6 READ	Tape = NULL	Store = NULL
State =	7 CONSULT	Tape = NULL	Store = NULL
State =	8 ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: aaabbbaa

State =	1 START	Tape = aaabbbaa	Store = NULL
State =	2 READ	Tape = aabbbaa	Store = NULL
State =	3 STORE	Tape = aabbbaa	Store = A
State =	4 READ	Tape = abbbaa	Store = A
State =	3 STORE	Tape = abbbaa	Store = AA
State =	4 READ	Tape = bbbaa	Store = AA
State =	3 STORE	Tape = bbbaa	Store = AAA
State =	4 READ	Tape = bbbaa	Store = AAA
State =	5 CONSULT	Tape = bbbaa	Store = AA
State =	6 READ	Tape = bbaa	Store = AA
State =	5 CONSULT	Tape = bbaa	Store = A
State =	6 READ	Tape = baa	Store = A
State =	5 CONSULT	Tape = baa	Store = NULL
State =	6 READ	Tape = aa	Store = NULL
State =	0 REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: abbbaabb

State =	1 START	Tape = abbbaabb	Store = NULL
State =	2 READ	Tape = bbaabb	Store = NULL
State =	3 STORE	Tape = bbaabb	Store = A
State =	4 READ	Tape = bbaabb	Store = A
State =	5 CONSULT	Tape = bbaabb	Store = NULL

```

State = 6 READ      Tape = aaabb  Store = NULL
State = 5 CONSULT   Tape = aaabb  Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaaaabbbbbbb

```

State = 1 START      Tape = aaaaaabbbbbbb  Store = NULL
State = 2 READ       Tape = aaaaaabbbbbbb  Store = NULL
State = 3 STORE      Tape = aaaaaabbbbbbb  Store = A
State = 4 READ       Tape = aaaabbbbbbb  Store = A
State = 3 STORE      Tape = aaaabbbbbbb  Store = AA
State = 4 READ       Tape = aaabbbbbbb  Store = AA
State = 3 STORE      Tape = aaabbbbbbb  Store = AAA
State = 4 READ       Tape = aabbbbbbb  Store = AAA
State = 3 STORE      Tape = aabbbbbbb  Store = AAAA
State = 4 READ       Tape = abbbbbbb  Store = AAAA
State = 3 STORE      Tape = abbbbbbb  Store = AAAAA
State = 4 READ       Tape = bbbbbbb  Store = AAAAA
State = 3 STORE      Tape = bbbbbbb  Store = AAAAAA
State = 4 READ       Tape = bbbbbb  Store = AAAAAA
State = 5 CONSULT    Tape = bbbbbb  Store = AAAAA
State = 6 READ       Tape = bbbb  Store = AAAAA
State = 5 CONSULT    Tape = bbbb  Store = AAAA
State = 6 READ       Tape = bbb  Store = AAAA
State = 5 CONSULT    Tape = bbb  Store = AAA
State = 6 READ       Tape = bb  Store = AAA
State = 5 CONSULT    Tape = bb  Store = AA
State = 6 READ       Tape = b  Store = AA
State = 5 CONSULT    Tape = b  Store = A
State = 6 READ       Tape = NULL  Store = A
State = 5 CONSULT    Tape = NULL  Store = NULL
State = 6 READ       Tape = NULL  Store = NULL
State = 7 CONSULT    Tape = NULL  Store = NULL
State = 8 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: abababbbaa

```

State = 1 START      Tape = abababbbaa  Store = NULL
State = 2 READ       Tape = bababbbaa  Store = NULL
State = 3 STORE      Tape = bababbbaa  Store = A
State = 4 READ       Tape = abababbbaa  Store = A
State = 5 CONSULT    Tape = abababbbaa  Store = NULL
State = 6 READ       Tape = babbbbaa  Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine:  
aaaaaaaaabbbbbbbbbb

```

State = 1 START      Tape = aaaaaaaaaabbbbbbbbbb  Store = NULL
State = 2 READ       Tape = aaaaaaaaaabbbbbbbbbb  Store = NULL
State = 3 STORE      Tape = aaaaaaaaaabbbbbbbbbb  Store = A
State = 4 READ       Tape = aaaaaaaaaabbbbbbbbbb  Store = A
State = 3 STORE      Tape = aaaaaaaaaabbbbbbbbbb  Store = AA
State = 4 READ       Tape = aaaaaaabbbbbbbbbb  Store = AA
State = 3 STORE      Tape = aaaaaaabbbbbbbbbb  Store = AAA
State = 4 READ       Tape = aaaaaaabbbbbbbbbb  Store = AAA
State = 3 STORE      Tape = aaaaaaabbbbbbbbbb  Store = AAAA
State = 4 READ       Tape = aaaaabbbbbbbbbb  Store = AAAA
State = 3 STORE      Tape = aaaaabbbbbbbbbb  Store = AAAAA
State = 4 READ       Tape = aaaabbbbbbbbbb  Store = AAAAA
State = 3 STORE      Tape = aaaabbbbbbbbbb  Store = AAAAAA
State = 4 READ       Tape = aaabbbbbbbbbb  Store = AAAAAA
State = 3 STORE      Tape = aaabbbbbbbbbb  Store = AAAAAA
State = 4 READ       Tape = aabbbbbbbbbb  Store = AAAAAA
State = 3 STORE      Tape = aabbbbbbbbbb  Store = AAAAAA

```

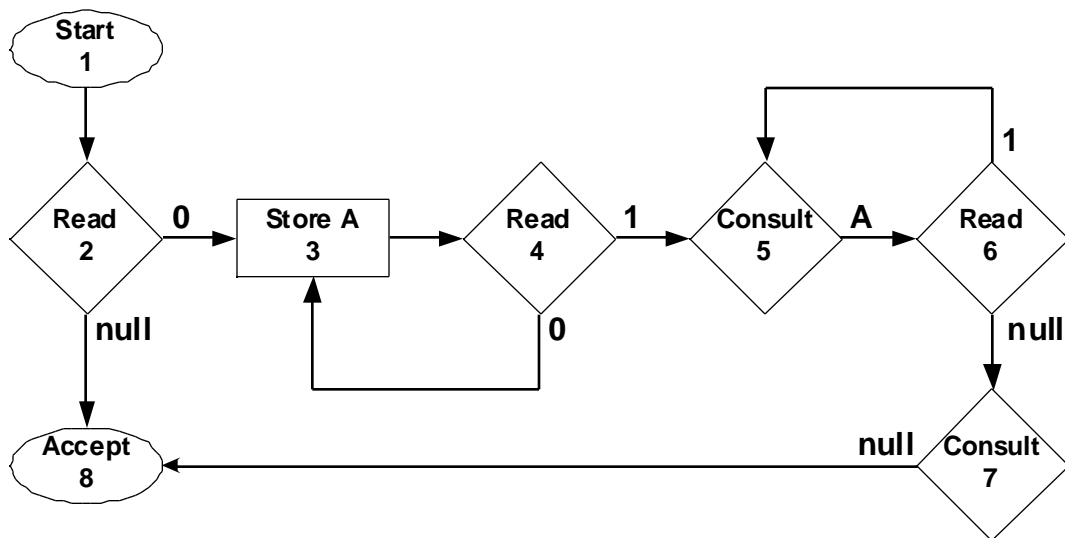




**Table 2. Post Machine describing the language { a<sup>n</sup>b<sup>n</sup> | n ≥ 0 }**

Note: input alphabet = {0, 1}

Number	State Type	Add Letter	Next State	Read/Branch			Consult/Branch		
				If 0	If 1	If null	If A	If B	If null
0	Reject								
1	Start		2						
2	Read			3	0	8			
3	Store	A	4						
4	Read	B	5						
5	Consult						6	0	0
6	Read			0	5	7			
7	Consult						0	0	8
8	Accept								



Menu:

1: Run Post Machine  
2: Quit

Enter: 1

Enter Tape alphabet:

Enter the size of the alphabet: 2  
Enter letter: 0  
Enter letter: 1

Alphabet:

Letter 1: 0

Letter 2: 1

Enter Store alphabet:

Enter the size of the alphabet: 2

Enter letter: A

Enter letter: B

Alphabet:

Letter 1: A

Letter 2: B

Defining Post Machine:

All Reject States are defined as 0 (zero).

Enter the number of states excluding rejected states: 8

State 1: START

Enter the next state (integer) : 2

State 2:

State Type:

1: Read Tape

2: Consult Queue

3: Store Queue

4: Shift Right Cyclically (SRC)

5: Add Front (AF)

6: Accept State

Enter a number: 1

Enter Next State (integer) :

    If 0: 3

    If 1: 0

    If NULL: 8

State 3:

State Type:

1: Read Tape

2: Consult Queue

3: Store Queue

4: Shift Right Cyclically (SRC)

5: Add Front (AF)

6: Accept State

Enter a number: 3

Enter letter to add at end of queue: A

Enter the next state (integer) : 4

State 4:

State Type:

1: Read Tape

2: Consult Queue

3: Store Queue

4: Shift Right Cyclically (SRC)

5: Add Front (AF)

6: Accept State

```

Enter a number: 1

Enter Next State (integer) :
    If 0: 3
    If 1: 5
    If NULL: 0

State 5:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 2

Enter Next State (integer) :
    If A: 6
    If B: 0
    If NULL: 0

State 6:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 1

Enter Next State (integer) :
    If 0: 0
    If 1: 5
    If NULL: 7

State 7:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 2

Enter Next State (integer) :
    If A: 0
    If B: 0
    If NULL: 8

State 8:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)

```

5: Add Front (AF)  
6: Accept State

Enter a number: 6

Number	State	Add	Queue	Next	If 0	If 1	If NULL	If A	If B	If NULL
0	X	-		0	0	0	0	0	0	0
1	I	-		2	0	0	0	0	0	0
2	R	-		0	3	0	8	0	0	0
3	W	A		4	0	0	0	0	0	0
4	R	B		5	3	5	0	0	0	0
5	K	-		0	0	0	0	6	0	0
6	R	-		0	0	5	7	0	0	0
7	K	-		0	0	0	0	0	0	8
8	A	-		0	0	0	0	0	0	9

Legend for State Type:

I = Start,      A = Accept,      X=Reject  
R = Read,      K = Consult,      W = Store,  
S=Shift Right Cyclically,      F = Add Front

Enter word or 'N' for null string or '\*' to turn off machine: N

State = 1 START      Tape = NULL      Store = NULL  
State = 2 READ      Tape = NULL      Store = NULL  
State = 8 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: 0

State = 1 START      Tape = 0      Store = NULL  
State = 2 READ      Tape = NULL      Store = NULL  
State = 3 STORE      Tape = NULL      Store = A  
State = 4 READ      Tape = NULL      Store = A  
State = 0 REJECT

Enter word or 'N' for null string or '\*' to turn off machine: 1

State = 1 START      Tape = 1      Store = NULL  
State = 2 READ      Tape = NULL      Store = NULL  
State = 0 REJECT

Enter word or 'N' for null string or '\*' to turn off machine: 01

State = 1 START      Tape = 01      Store = NULL  
State = 2 READ      Tape = 1      Store = NULL  
State = 3 STORE      Tape = 1      Store = A  
State = 4 READ      Tape = NULL      Store = A  
State = 5 CONSULT      Tape = NULL      Store = NULL  
State = 6 READ      Tape = NULL      Store = NULL  
State = 7 CONSULT      Tape = NULL      Store = NULL  
State = 8 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: 010

State = 1 START      Tape = 010      Store = NULL  
State = 2 READ      Tape = 10      Store = NULL  
State = 3 STORE      Tape = 10      Store = A  
State = 4 READ      Tape = 0      Store = A  
State = 5 CONSULT      Tape = 0      Store = NULL  
State = 6 READ      Tape = NULL      Store = NULL  
State = 0 REJECT

Enter word or 'N' for null string or '\*' to turn off machine: 000111

```

State = 1 START      Tape = 000111      Store = NULL
State = 2 READ       Tape = 00111      Store = NULL
State = 3 STORE      Tape = 00111      Store = A
State = 4 READ       Tape = 0111       Store = A
State = 3 STORE      Tape = 0111       Store = AA
State = 4 READ       Tape = 111        Store = AA
State = 3 STORE      Tape = 111        Store = AAA
State = 4 READ       Tape = 11         Store = AAA
State = 5 CONSULT    Tape = 11         Store = AA
State = 6 READ       Tape = 1          Store = AA
State = 5 CONSULT    Tape = 1          Store = A
State = 6 READ       Tape = NULL       Store = A
State = 5 CONSULT    Tape = NULL       Store = NULL
State = 6 READ       Tape = NULL       Store = NULL
State = 7 CONSULT    Tape = NULL       Store = NULL
State = 8 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: 111000

```

State = 1 START      Tape = 111000     Store = NULL
State = 2 READ       Tape = 11000      Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: 000000111111

```

State = 1 START      Tape = 000000111111 Store = NULL
State = 2 READ       Tape = 00000111111  Store = NULL
State = 3 STORE      Tape = 00000111111  Store = A
State = 4 READ       Tape = 0000111111   Store = A
State = 3 STORE      Tape = 0000111111   Store = AA
State = 4 READ       Tape = 000111111    Store = AA
State = 3 STORE      Tape = 000111111    Store = AAA
State = 4 READ       Tape = 00111111     Store = AAA
State = 3 STORE      Tape = 00111111     Store = AAAA
State = 4 READ       Tape = 0111111      Store = AAAA
State = 3 STORE      Tape = 0111111      Store = AAAAA
State = 4 READ       Tape = 111111       Store = AAAAA
State = 3 STORE      Tape = 111111       Store = AAAAAA
State = 4 READ       Tape = 11111        Store = AAAAAA
State = 5 CONSULT    Tape = 11111        Store = AAAAA
State = 6 READ       Tape = 1111         Store = AAAAA
State = 5 CONSULT    Tape = 1111         Store = AAAA
State = 6 READ       Tape = 111          Store = AAAA
State = 5 CONSULT    Tape = 111          Store = AAA
State = 6 READ       Tape = 11           Store = AAA
State = 5 CONSULT    Tape = 11           Store = AA
State = 6 READ       Tape = 1            Store = AA
State = 5 CONSULT    Tape = 1            Store = A
State = 6 READ       Tape = NULL         Store = A
State = 5 CONSULT    Tape = NULL         Store = NULL
State = 6 READ       Tape = NULL         Store = NULL
State = 7 CONSULT    Tape = NULL         Store = NULL
State = 8 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: 0011001100

```

State = 1 START      Tape = 0011001100   Store = NULL
State = 2 READ       Tape = 011001100    Store = NULL
State = 3 STORE      Tape = 011001100    Store = A
State = 4 READ       Tape = 11001100     Store = A
State = 3 STORE      Tape = 11001100     Store = AA
State = 4 READ       Tape = 1001100      Store = AA
State = 5 CONSULT    Tape = 1001100      Store = A
State = 6 READ       Tape = 001100       Store = A

```

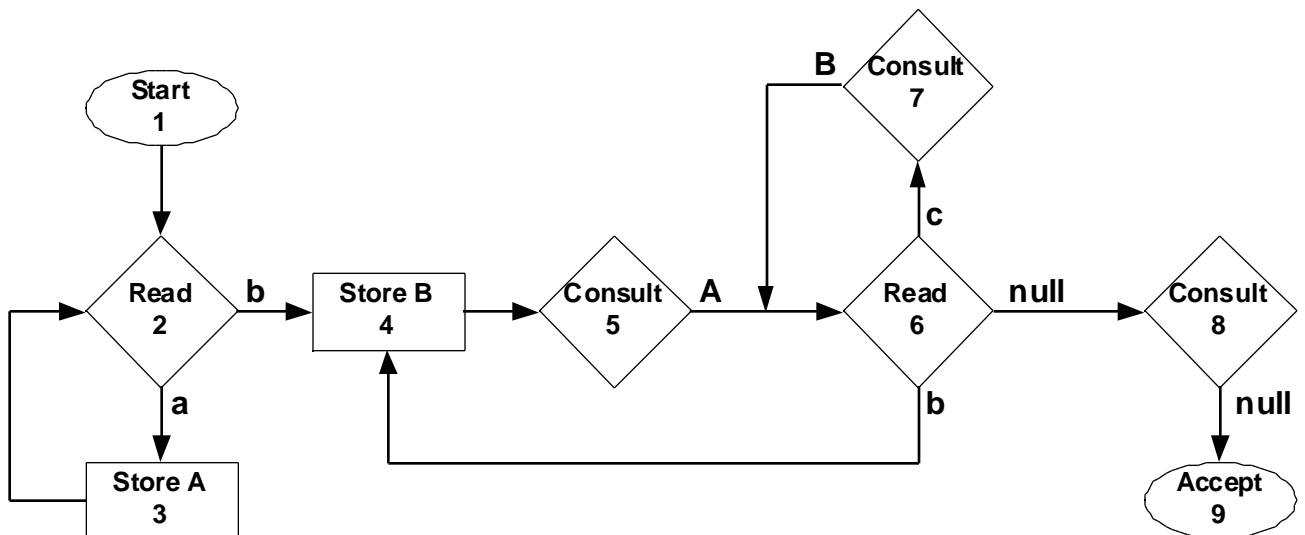


State = 0 REJECT

-----a<sup>n</sup>b<sup>n</sup>c<sup>n</sup>-----

**Table 3. Post Machine describing the language { a<sup>n</sup>b<sup>n</sup>c<sup>n</sup> | n > 0 }**

Number	State Type	Add Letter	Next State	Read/Branch				Consult/Branch		
				If a	If b	If c	If null	If A	If B	If null
0	Reject									
1	Start		2							
2	Read			3	4	0	0			
3	Store	A	2							
4	Store	B	5							
5	Consult							6	0	0
6	Read			0	4	7	8			
7	Consult							0	6	0
8	Consult							0	0	9
9	Accept									



Menu:

1: Run Post Machine  
2: Quit

Enter: 1

Enter Tape alphabet:

Enter the size of the alphabet: 3

Enter letter: a

Enter letter: b

Enter letter: c

Alphabet:

Letter 1: a

Letter 2: b

Letter 3: c



```

Enter Store alphabet:

Enter the size of the alphabet: 2
Enter letter: A
Enter letter: B

Alphabet:
Letter 1: A
Letter 2: B

Defining Post Machine:

All Reject States are defined as 0 (zero).

Enter the number of states excluding rejected states: 9

State 1: START

Enter the next state (integer) : 2

State 2:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 1

Enter Next State (integer) :
    If a: 3
    If b: 4
    If c: 0
    If NULL: 0

State 3:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 3
Enter letter to add at end of queue: A

Enter the next state (integer) : 2

State 4:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 3

```

Enter letter to add at end of queue: B

Enter the next state (integer) : 5

State 5:

State Type:

- 1: Read Tape
- 2: Consult Queue
- 3: Store Queue
- 4: Shift Right Cyclically (SRC)
- 5: Add Front (AF)
- 6: Accept State

Enter a number: 2

Enter Next State (integer) :

- If A: 6
- If B: 0
- If NULL: 0

State 6:

State Type:

- 1: Read Tape
- 2: Consult Queue
- 3: Store Queue
- 4: Shift Right Cyclically (SRC)
- 5: Add Front (AF)
- 6: Accept State

Enter a number: 1

Enter Next State (integer) :

- If a: 0
- If b: 4
- If c: 7
- If NULL: 8

State 7:

State Type:

- 1: Read Tape
- 2: Consult Queue
- 3: Store Queue
- 4: Shift Right Cyclically (SRC)
- 5: Add Front (AF)
- 6: Accept State

Enter a number: 2

Enter Next State (integer) :

- If A: 0
- If B: 6
- If NULL: 0

State 8:

State Type:

- 1: Read Tape
- 2: Consult Queue
- 3: Store Queue
- 4: Shift Right Cyclically (SRC)
- 5: Add Front (AF)
- 6: Accept State

Enter a number: 2

Enter Next State (integer) :

If A: 0

If B: 0

If NULL: 9

State 9:

State Type:

1: Read Tape

2: Consult Queue

3: Store Queue

4: Shift Right Cyclically (SRC)

5: Add Front (AF)

6: Accept State

Enter a number: 6

Number	State	Add	Queue	Next	If a	If b	If c	If NULL	If A	If B	If NULL
0	X	-		0	0	0	0	0	0	0	0
1	I	-		2	0	0	0	0	0	0	0
2	R	-		0	3	4	0	0	0	0	0
3	W	A		2	0	0	0	0	0	0	0
4	W	B		5	3	5	0	0	0	0	0
5	K	-		0	0	0	0	0	6	0	0
6	R	-		0	0	4	7	8	0	0	0
7	K	-		0	0	0	0	0	0	6	0
8	K	-		0	0	0	0	0	0	0	9
9	A	-		0	0	0	0	0	0	0	0

Legend for State Type:

I = Start, A = Accept, X=Reject

R = Read, K = Consult, W = Store,

S=Shift Right Cyclically, F = Add Front

Enter word or 'N' for null string or '\*' to turn off machine: N

State = 1 START      Tape = NULL      Store = NULL  
State = 2 READ      Tape = NULL      Store = NULL  
State = 0 REJECT

Enter word or 'N' for null string or '\*' to turn off machine: a

State = 1 START      Tape = a      Store = NULL  
State = 2 READ      Tape = NULL      Store = NULL  
State = 3 STORE      Tape = NULL      Store = A  
State = 2 READ      Tape = NULL      Store = A  
State = 0 REJECT

Enter word or 'N' for null string or '\*' to turn off machine: bac

State = 1 START      Tape = bac      Store = NULL  
State = 2 READ      Tape = ac      Store = NULL  
State = 4 STORE      Tape = ac      Store = B  
State = 5 CONSULT      Tape = ac      Store = NULL  
State = 0 REJECT

Enter word or 'N' for null string or '\*' to turn off machine: abc

State = 1 START      Tape = abc      Store = NULL  
State = 2 READ      Tape = bc      Store = NULL

State = 3	STORE	Tape = bc	Store = A
State = 2	READ	Tape = c	Store = A
State = 4	STORE	Tape = c	Store = AB
State = 5	CONSULT	Tape = c	Store = B
State = 6	READ	Tape = NULL	Store = B
State = 7	CONSULT	Tape = NULL	Store = NULL
State = 6	READ	Tape = NULL	Store = NULL
State = 8	CONSULT	Tape = NULL	Store = NULL
State = 9	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: aabbbcc

State = 1	START	Tape = aabbbcc	Store = NULL
State = 2	READ	Tape = abbbcc	Store = NULL
State = 3	STORE	Tape = abbbcc	Store = A
State = 2	READ	Tape = bbbcc	Store = A
State = 3	STORE	Tape = bbbcc	Store = AA
State = 2	READ	Tape = bbcc	Store = AA
State = 4	STORE	Tape = bbcc	Store = AAB
State = 5	CONSULT	Tape = bbcc	Store = AB
State = 6	READ	Tape = bcc	Store = AB
State = 4	STORE	Tape = bcc	Store = ABB
State = 5	CONSULT	Tape = bcc	Store = BB
State = 6	READ	Tape = cc	Store = BB
State = 4	STORE	Tape = cc	Store = BBB
State = 5	CONSULT	Tape = cc	Store = BB
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: aaabbccc

State = 1	START	Tape = aaabbccc	Store = NULL
State = 2	READ	Tape = aabbccc	Store = NULL
State = 3	STORE	Tape = aabbccc	Store = A
State = 2	READ	Tape = abbccc	Store = A
State = 3	STORE	Tape = abbccc	Store = AA
State = 2	READ	Tape = bbccc	Store = AA
State = 3	STORE	Tape = bbccc	Store = AAA
State = 2	READ	Tape = bccc	Store = AAA
State = 4	STORE	Tape = bccc	Store = AAAB
State = 5	CONSULT	Tape = bccc	Store = AAB
State = 6	READ	Tape = ccc	Store = AAB
State = 4	STORE	Tape = ccc	Store = AABB
State = 5	CONSULT	Tape = ccc	Store = ABB
State = 6	READ	Tape = cc	Store = ABB
State = 7	CONSULT	Tape = cc	Store = BB
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: aabbccccc

State = 1	START	Tape = aabbccccc	Store = NULL
State = 2	READ	Tape = abbccccc	Store = NULL
State = 3	STORE	Tape = abbccccc	Store = A
State = 2	READ	Tape = bbccccc	Store = A
State = 3	STORE	Tape = bbccccc	Store = AA
State = 2	READ	Tape = bccccc	Store = AA
State = 4	STORE	Tape = bccccc	Store = AAB
State = 5	CONSULT	Tape = bccccc	Store = AB
State = 6	READ	Tape = cccc	Store = AB
State = 4	STORE	Tape = cccc	Store = ABB
State = 5	CONSULT	Tape = cccc	Store = BB
State = 6	READ	Tape = ccc	Store = BB
State = 7	CONSULT	Tape = ccc	Store = B
State = 6	READ	Tape = cc	Store = B
State = 7	CONSULT	Tape = cc	Store = NULL
State = 6	READ	Tape = c	Store = NULL

```

State = 7 CONSULT    Tape = c    Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaccbb

```

State = 1 START      Tape = aaccbb    Store = NULL
State = 2 READ       Tape = accbb     Store = NULL
State = 3 STORE      Tape = accbb     Store = A
State = 2 READ       Tape = ccbb      Store = A
State = 3 STORE      Tape = ccbb      Store = AA
State = 2 READ       Tape = cbb       Store = AA
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbaacc

```

State = 1 START      Tape = bbaacc    Store = NULL
State = 2 READ       Tape = baacc     Store = NULL
State = 4 STORE      Tape = baacc     Store = B
State = 5 CONSULT    Tape = baacc     Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaabbbbbaaaa

```

State = 1 START      Tape = aaaabbbbbaaaa    Store = NULL
State = 2 READ       Tape = aaabbbbbaaaa    Store = NULL
State = 3 STORE      Tape = aaabbbbbaaaa    Store = A
State = 2 READ       Tape = aabbbbbaaaa     Store = A
State = 3 STORE      Tape = aabbbbbaaaa     Store = AA
State = 2 READ       Tape = abbbbbaaaa      Store = AA
State = 3 STORE      Tape = abbbbbaaaa      Store = AAA
State = 2 READ       Tape = bbbbbaaaa       Store = AAA
State = 3 STORE      Tape = bbbbbaaaa       Store = AAAA
State = 2 READ       Tape = bbbbaaaa        Store = AAAA
State = 4 STORE      Tape = bbbbaaaa        Store = AAAAB
State = 5 CONSULT    Tape = bbbbaaaa        Store = AAAB
State = 6 READ       Tape = bbbaaaa         Store = AAAB
State = 4 STORE      Tape = bbbaaaa         Store = AAABB
State = 5 CONSULT    Tape = bbbaaaa         Store = AABBB
State = 6 READ       Tape = baaaa          Store = AABBB
State = 4 STORE      Tape = baaaa          Store = AABBBB
State = 5 CONSULT    Tape = baaaa          Store = ABBBB
State = 6 READ       Tape = aaaa           Store = ABBBB
State = 4 STORE      Tape = aaaa           Store = ABBBBB
State = 5 CONSULT    Tape = aaaa           Store = BBBB
State = 6 READ       Tape = aaa            Store = BBBB
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaaabbbbcccccc

```

State = 1 START      Tape = aaaaabbbbcccccc    Store = NULL
State = 2 READ       Tape = aaaaabbbbcccccc    Store = NULL
State = 3 STORE      Tape = aaaaabbbbcccccc    Store = A
State = 2 READ       Tape = aaabbbbcccccc     Store = A
State = 3 STORE      Tape = aaabbbbcccccc     Store = AA
State = 2 READ       Tape = aabbbbcccccc      Store = AA
State = 3 STORE      Tape = aabbbbcccccc      Store = AAA
State = 2 READ       Tape = abbbbcccccc       Store = AAA
State = 3 STORE      Tape = abbbbcccccc       Store = AAAA
State = 2 READ       Tape = bbbbcccccc        Store = AAAA
State = 3 STORE      Tape = bbbbcccccc        Store = AAAAA
State = 2 READ       Tape = bbbbcccccc        Store = AAAAA
State = 4 STORE      Tape = bbbbcccccc        Store = AAAAAAB
State = 5 CONSULT    Tape = bbbbcccccc        Store = AAAAAAB
State = 6 READ       Tape = bbbbcccccc        Store = AAAAAAB
State = 4 STORE      Tape = bbbbcccccc        Store = AAAAAAB

```

```
Enter word or 'N' for null string or '*' to turn off machine:
aaaaaaaaabbbbbbbccccccc
```

102

State = 6	READ	Tape = ccccccc	Store = ABBBBBBB
State = 4	STORE	Tape = ccccccc	Store = ABBBBBBBBB
State = 5	CONSULT	Tape = ccccccc	Store = BBBBBBBB
State = 6	READ	Tape = ccccccc	Store = BBBBBBBB
State = 7	CONSULT	Tape = ccccccc	Store = BBBBBBB
State = 6	READ	Tape = ccccccc	Store = BBBBBBB
State = 7	CONSULT	Tape = ccccccc	Store = BBBBBB
State = 6	READ	Tape = ccccc	Store = BBBBBB
State = 7	CONSULT	Tape = ccccc	Store = BBBBBB
State = 6	READ	Tape = ccccc	Store = BBBBBB
State = 7	CONSULT	Tape = ccccc	Store = BBBBB
State = 6	READ	Tape = cccc	Store = BBBBB
State = 7	CONSULT	Tape = cccc	Store = BBBB
State = 6	READ	Tape = ccc	Store = BBBB
State = 7	CONSULT	Tape = ccc	Store = BBB
State = 6	READ	Tape = cc	Store = BBB
State = 7	CONSULT	Tape = cc	Store = BB
State = 6	READ	Tape = c	Store = BB
State = 7	CONSULT	Tape = c	Store = B
State = 6	READ	Tape = NULL	Store = B
State = 7	CONSULT	Tape = NULL	Store = NULL
State = 6	READ	Tape = NULL	Store = NULL
State = 8	CONSULT	Tape = NULL	Store = NULL
State = 9	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: aabb

State = 1	START	Tape = aabb	Store = NULL
State = 2	READ	Tape = abb	Store = NULL
State = 3	STORE	Tape = abb	Store = A
State = 2	READ	Tape = bb	Store = A
State = 3	STORE	Tape = bb	Store = AA
State = 2	READ	Tape = b	Store = AA
State = 4	STORE	Tape = b	Store = AAB
State = 5	CONSULT	Tape = b	Store = AB
State = 6	READ	Tape = NULL	Store = AB
State = 4	STORE	Tape = NULL	Store = ABB
State = 5	CONSULT	Tape = NULL	Store = BB
State = 6	READ	Tape = NULL	Store = BB
State = 8	CONSULT	Tape = NULL	Store = B
State = 0	REJECT		

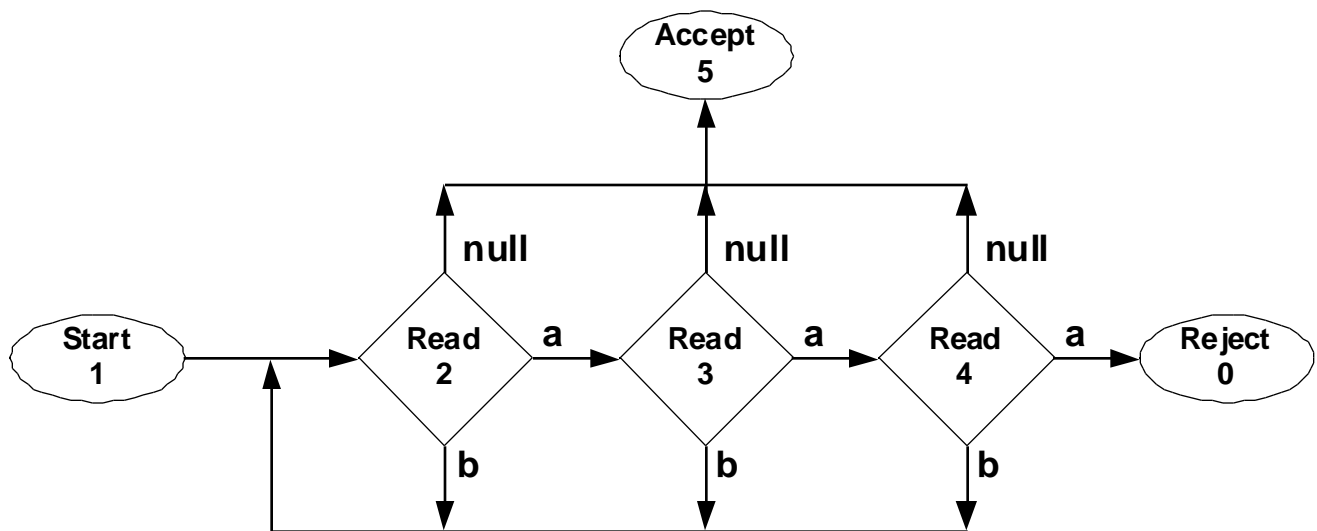
Enter word or 'N' for null string or '\*' to turn off machine: aaccbb

State = 1	START	Tape = aaccbb	Store = NULL
State = 2	READ	Tape = accbb	Store = NULL
State = 3	STORE	Tape = accbb	Store = A
State = 2	READ	Tape = cbb	Store = A
State = 3	STORE	Tape = cbb	Store = AA
State = 2	READ	Tape = cbb	Store = AA
State = 0	REJECT		

-----notaaa-----

**Table 4. Post Machine describing the language no sub-strings of aaa**

Number	State Type	Add Letter	Next State	Read/Branch		
				If a	If b	If null
0	Reject					
1	Start		2			
2	Read			3	2	5
3	Read			4	2	5
4	Read			0	2	5
5	Accept					



Menu:

1: Run Post Machine  
2: Quit

Enter: 1

Enter Tape alphabet:

Enter the size of the alphabet: 2  
Enter letter: a  
Enter letter: b

Alphabet:  
Letter 1: a  
Letter 2: b

Enter Store alphabet:

If not applicable, Enter 0 (zero)

Enter the size of the alphabet: 0

Defining Post Machine:



All Reject States are defined as 0 (zero).

Enter the number of states excluding rejected states: 5

State 1: START

Enter the next state (integer) : 2

State 2:

State Type:

1: Read Tape  
2: Consult Queue  
3: Store Queue  
4: Shift Right Cyclically (SRC)  
5: Add Front (AF)  
6: Accept State

Enter a number: 1

Enter Next State (integer) :

If a: 3  
If b: 2  
If NULL: 5

State 3:

State Type:

1: Read Tape  
2: Consult Queue  
3: Store Queue  
4: Shift Right Cyclically (SRC)  
5: Add Front (AF)  
6: Accept State

Enter a number: 1

Enter Next State (integer) :

If a: 4  
If b: 2  
If NULL: 5

State 4:

State Type:

1: Read Tape  
2: Consult Queue  
3: Store Queue  
4: Shift Right Cyclically (SRC)  
5: Add Front (AF)  
6: Accept State

Enter a number: 1

Enter Next State (integer) :

If a: 0  
If b: 2  
If NULL: 5

State 5:

State Type:

1: Read Tape  
2: Consult Queue  
3: Store Queue

3: Store Queue  
 4: Shift Right Cyclically (SRC)  
 5: Add Front (AF)  
 6: Accept State

Enter a number: 6

Print Transition Table:

Number	State	Add	Queue	Next	If a	If b	If NULL
0	X	-		0	0	0	0
1	I	-		2	0	0	0
2	R	-		0	3	2	5
3	R	-		0	4	2	5
4	R	-		0	0	2	5
5	A	-		0	0	0	0

Legend for State Type:

I = Start,      A = Accept,      X=Reject  
 R = Read,      K = Consult,      W = Store,  
 S=Shift Right Cyclically,      F = Add Front

Enter word or 'N' for null string or '\*' to turn off machine: N

State = 1 START      Tape = NULL      Store = NULL  
 State = 2 READ      Tape = NULL      Store = NULL  
 State = 5 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: a

State = 1 START      Tape = a      Store = NULL  
 State = 2 READ      Tape = NULL      Store = NULL  
 State = 3 READ      Tape = NULL      Store = NULL  
 State = 5 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: b

State = 1 START      Tape = b      Store = NULL  
 State = 2 READ      Tape = NULL      Store = NULL  
 State = 2 READ      Tape = NULL      Store = NULL  
 State = 5 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: aa

State = 1 START      Tape = aa      Store = NULL  
 State = 2 READ      Tape = a      Store = NULL  
 State = 3 READ      Tape = NULL      Store = NULL  
 State = 4 READ      Tape = NULL      Store = NULL  
 State = 5 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: ab

State = 1 START      Tape = ab      Store = NULL  
 State = 2 READ      Tape = b      Store = NULL  
 State = 3 READ      Tape = NULL      Store = NULL  
 State = 2 READ      Tape = NULL      Store = NULL  
 State = 5 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: ba

State = 1 START      Tape = ba      Store = NULL  
 State = 2 READ      Tape = a      Store = NULL  
 State = 2 READ      Tape = NULL      Store = NULL

State = 3 READ           Tape = NULL       Store = NULL  
State = 5 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: bb

State = 1 START        Tape = bb       Store = NULL  
State = 2 READ        Tape = b       Store = NULL  
State = 2 READ        Tape = NULL       Store = NULL  
State = 2 READ        Tape = NULL       Store = NULL  
State = 5 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: aaa

State = 1 START        Tape = aaa       Store = NULL  
State = 2 READ        Tape = aa       Store = NULL  
State = 3 READ        Tape = a       Store = NULL  
State = 4 READ        Tape = NULL       Store = NULL  
State = 0 REJECT

Enter word or 'N' for null string or '\*' to turn off machine: aab

State = 1 START        Tape = aab       Store = NULL  
State = 2 READ        Tape = ab       Store = NULL  
State = 3 READ        Tape = b       Store = NULL  
State = 4 READ        Tape = NULL       Store = NULL  
State = 2 READ        Tape = NULL       Store = NULL  
State = 5 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: aba

State = 1 START        Tape = aba       Store = NULL  
State = 2 READ        Tape = ba       Store = NULL  
State = 3 READ        Tape = a       Store = NULL  
State = 2 READ        Tape = NULL       Store = NULL  
State = 3 READ        Tape = NULL       Store = NULL  
State = 5 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: abb

State = 1 START        Tape = abb       Store = NULL  
State = 2 READ        Tape = bb       Store = NULL  
State = 3 READ        Tape = b       Store = NULL  
State = 2 READ        Tape = NULL       Store = NULL  
State = 2 READ        Tape = NULL       Store = NULL  
State = 5 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: baa

State = 1 START        Tape = baa       Store = NULL  
State = 2 READ        Tape = aa       Store = NULL  
State = 2 READ        Tape = a       Store = NULL  
State = 3 READ        Tape = NULL       Store = NULL  
State = 4 READ        Tape = NULL       Store = NULL  
State = 5 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: bab

State = 1 START        Tape = bab       Store = NULL  
State = 2 READ        Tape = ab       Store = NULL  
State = 2 READ        Tape = b       Store = NULL  
State = 3 READ        Tape = NULL       Store = NULL  
State = 2 READ        Tape = NULL       Store = NULL  
State = 5 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: bba

```

State = 1 START      Tape = bba      Store = NULL
State = 2 READ       Tape = ba       Store = NULL
State = 2 READ       Tape = a        Store = NULL
State = 2 READ       Tape = NULL     Store = NULL
State = 3 READ       Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbb

```

State = 1 START      Tape = bbb      Store = NULL
State = 2 READ       Tape = bb       Store = NULL
State = 2 READ       Tape = b        Store = NULL
State = 2 READ       Tape = NULL     Store = NULL
State = 2 READ       Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaa

```

State = 1 START      Tape = aaaa     Store = NULL
State = 2 READ       Tape = aaa      Store = NULL
State = 3 READ       Tape = aa       Store = NULL
State = 4 READ       Tape = a        Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaab

```

State = 1 START      Tape = aaab     Store = NULL
State = 2 READ       Tape = aab      Store = NULL
State = 3 READ       Tape = ab       Store = NULL
State = 4 READ       Tape = b        Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaba

```

State = 1 START      Tape = aaba     Store = NULL
State = 2 READ       Tape = aba      Store = NULL
State = 3 READ       Tape = ba       Store = NULL
State = 4 READ       Tape = a        Store = NULL
State = 2 READ       Tape = NULL     Store = NULL
State = 3 READ       Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aabb

```

State = 1 START      Tape = aabb     Store = NULL
State = 2 READ       Tape = abb      Store = NULL
State = 3 READ       Tape = bb       Store = NULL
State = 4 READ       Tape = b        Store = NULL
State = 2 READ       Tape = NULL     Store = NULL
State = 2 READ       Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: abaa

```

State = 1 START      Tape = abaa     Store = NULL
State = 2 READ       Tape = baa      Store = NULL
State = 3 READ       Tape = aa       Store = NULL
State = 2 READ       Tape = a        Store = NULL
State = 3 READ       Tape = NULL     Store = NULL
State = 4 READ       Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: abab

```

State = 1 START      Tape = abab      Store = NULL
State = 2 READ       Tape = bab       Store = NULL
State = 3 READ       Tape = ab        Store = NULL
State = 2 READ       Tape = b         Store = NULL
State = 3 READ       Tape = NULL      Store = NULL
State = 2 READ       Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: abba

```

State = 1 START      Tape = abba      Store = NULL
State = 2 READ       Tape = bba       Store = NULL
State = 3 READ       Tape = ba        Store = NULL
State = 2 READ       Tape = a         Store = NULL
State = 2 READ       Tape = NULL      Store = NULL
State = 3 READ       Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: abbb

```

State = 1 START      Tape = abbb      Store = NULL
State = 2 READ       Tape = bbb       Store = NULL
State = 3 READ       Tape = bb        Store = NULL
State = 2 READ       Tape = b         Store = NULL
State = 2 READ       Tape = NULL      Store = NULL
State = 2 READ       Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: baaa

```

State = 1 START      Tape = baaa      Store = NULL
State = 2 READ       Tape = aaa       Store = NULL
State = 2 READ       Tape = aa        Store = NULL
State = 3 READ       Tape = a         Store = NULL
State = 4 READ       Tape = NULL      Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: baab

```

State = 1 START      Tape = baab      Store = NULL
State = 2 READ       Tape = aab       Store = NULL
State = 2 READ       Tape = ab        Store = NULL
State = 3 READ       Tape = b         Store = NULL
State = 4 READ       Tape = NULL      Store = NULL
State = 2 READ       Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: baba

```

State = 1 START      Tape = baba      Store = NULL
State = 2 READ       Tape = aba       Store = NULL
State = 2 READ       Tape = ba        Store = NULL
State = 3 READ       Tape = a         Store = NULL
State = 2 READ       Tape = NULL      Store = NULL
State = 3 READ       Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: babb

```

State = 1 START      Tape = babb      Store = NULL
State = 2 READ       Tape = abb       Store = NULL
State = 2 READ       Tape = bb        Store = NULL
State = 3 READ       Tape = b         Store = NULL
State = 2 READ       Tape = NULL      Store = NULL
State = 2 READ       Tape = NULL      Store = NULL

```

State = 5 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: bbba

State = 1	START	Tape = bbba	Store = NULL
State = 2	READ	Tape = bba	Store = NULL
State = 2	READ	Tape = aa	Store = NULL
State = 2	READ	Tape = a	Store = NULL
State = 3	READ	Tape = NULL	Store = NULL
State = 4	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: bbab

State = 1	START	Tape = bbab	Store = NULL
State = 2	READ	Tape = bab	Store = NULL
State = 2	READ	Tape = ab	Store = NULL
State = 2	READ	Tape = b	Store = NULL
State = 3	READ	Tape = NULL	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: bbba

State = 1	START	Tape = bbba	Store = NULL
State = 2	READ	Tape = bba	Store = NULL
State = 2	READ	Tape = ba	Store = NULL
State = 2	READ	Tape = a	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 3	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: bbbb

State = 1	START	Tape = bbbb	Store = NULL
State = 2	READ	Tape = bbb	Store = NULL
State = 2	READ	Tape = bb	Store = NULL
State = 2	READ	Tape = b	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: aaaaa

State = 1	START	Tape = aaaaa	Store = NULL
State = 2	READ	Tape = aaaa	Store = NULL
State = 3	READ	Tape = aaa	Store = NULL
State = 4	READ	Tape = aa	Store = NULL
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: aaaab

State = 1	START	Tape = aaaab	Store = NULL
State = 2	READ	Tape = aaab	Store = NULL
State = 3	READ	Tape = aab	Store = NULL
State = 4	READ	Tape = ab	Store = NULL
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: aaaba

State = 1	START	Tape = aaaba	Store = NULL
State = 2	READ	Tape = aaba	Store = NULL
State = 3	READ	Tape = aba	Store = NULL
State = 4	READ	Tape = ba	Store = NULL
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: aaabb

State =	1	START	Tape =	aaabb	Store =	NULL
State =	2	READ	Tape =	aabb	Store =	NULL
State =	3	READ	Tape =	abb	Store =	NULL
State =	4	READ	Tape =	bb	Store =	NULL
State =	0	REJECT				

Enter word or 'N' for null string or '\*' to turn off machine: aabaa

State =	1	START	Tape =	aabaa	Store =	NULL
State =	2	READ	Tape =	abaa	Store =	NULL
State =	3	READ	Tape =	baa	Store =	NULL
State =	4	READ	Tape =	aa	Store =	NULL
State =	2	READ	Tape =	a	Store =	NULL
State =	3	READ	Tape =	NULL	Store =	NULL
State =	4	READ	Tape =	NULL	Store =	NULL
State =	5	ACCEPT				

Enter word or 'N' for null string or '\*' to turn off machine: aabab

State =	1	START	Tape =	aabab	Store =	NULL
State =	2	READ	Tape =	abab	Store =	NULL
State =	3	READ	Tape =	bab	Store =	NULL
State =	4	READ	Tape =	ab	Store =	NULL
State =	2	READ	Tape =	b	Store =	NULL
State =	3	READ	Tape =	NULL	Store =	NULL
State =	2	READ	Tape =	NULL	Store =	NULL
State =	5	ACCEPT				

Enter word or 'N' for null string or '\*' to turn off machine: aabba

State =	1	START	Tape =	aabba	Store =	NULL
State =	2	READ	Tape =	abba	Store =	NULL
State =	3	READ	Tape =	bba	Store =	NULL
State =	4	READ	Tape =	ba	Store =	NULL
State =	2	READ	Tape =	a	Store =	NULL
State =	2	READ	Tape =	NULL	Store =	NULL
State =	3	READ	Tape =	NULL	Store =	NULL
State =	5	ACCEPT				

Enter word or 'N' for null string or '\*' to turn off machine: aabbb

State =	1	START	Tape =	aabbb	Store =	NULL
State =	2	READ	Tape =	abbb	Store =	NULL
State =	3	READ	Tape =	bbb	Store =	NULL
State =	4	READ	Tape =	bb	Store =	NULL
State =	2	READ	Tape =	b	Store =	NULL
State =	2	READ	Tape =	NULL	Store =	NULL
State =	2	READ	Tape =	NULL	Store =	NULL
State =	5	ACCEPT				

Enter word or 'N' for null string or '\*' to turn off machine: abaaa

State =	1	START	Tape =	abaaa	Store =	NULL
State =	2	READ	Tape =	baaa	Store =	NULL
State =	3	READ	Tape =	aaa	Store =	NULL
State =	2	READ	Tape =	aa	Store =	NULL
State =	3	READ	Tape =	a	Store =	NULL
State =	4	READ	Tape =	NULL	Store =	NULL
State =	0	REJECT				

Enter word or 'N' for null string or '\*' to turn off machine: abaab

State = 1	START	Tape = abaab	Store = NULL
State = 2	READ	Tape = baab	Store = NULL
State = 3	READ	Tape = aab	Store = NULL
State = 2	READ	Tape = ab	Store = NULL
State = 3	READ	Tape = b	Store = NULL
State = 4	READ	Tape = NULL	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: ababa

State = 1	START	Tape = ababa	Store = NULL
State = 2	READ	Tape = baba	Store = NULL
State = 3	READ	Tape = aba	Store = NULL
State = 2	READ	Tape = ba	Store = NULL
State = 3	READ	Tape = a	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 3	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: ababb

State = 1	START	Tape = ababb	Store = NULL
State = 2	READ	Tape = babb	Store = NULL
State = 3	READ	Tape = abb	Store = NULL
State = 2	READ	Tape = bb	Store = NULL
State = 3	READ	Tape = b	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: abbba

State = 1	START	Tape = abbba	Store = NULL
State = 2	READ	Tape = bbba	Store = NULL
State = 3	READ	Tape = bba	Store = NULL
State = 2	READ	Tape = aa	Store = NULL
State = 2	READ	Tape = a	Store = NULL
State = 3	READ	Tape = NULL	Store = NULL
State = 4	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: abbab

State = 1	START	Tape = abbab	Store = NULL
State = 2	READ	Tape = bbab	Store = NULL
State = 3	READ	Tape = bab	Store = NULL
State = 2	READ	Tape = ab	Store = NULL
State = 2	READ	Tape = b	Store = NULL
State = 3	READ	Tape = NULL	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: abbba

State = 1	START	Tape = abbba	Store = NULL
State = 2	READ	Tape = bbba	Store = NULL
State = 3	READ	Tape = bba	Store = NULL
State = 2	READ	Tape = ba	Store = NULL
State = 2	READ	Tape = a	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 3	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: abbbb



State = 1	START	Tape = abbbb	Store = NULL
State = 2	READ	Tape = bbbb	Store = NULL
State = 3	READ	Tape = bbb	Store = NULL
State = 2	READ	Tape = bb	Store = NULL
State = 2	READ	Tape = b	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: baaaa

State = 1	START	Tape = baaaa	Store = NULL
State = 2	READ	Tape = aaaa	Store = NULL
State = 2	READ	Tape = aaa	Store = NULL
State = 3	READ	Tape = aa	Store = NULL
State = 4	READ	Tape = a	Store = NULL
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: baaab

State = 1	START	Tape = baaab	Store = NULL
State = 2	READ	Tape = aaab	Store = NULL
State = 2	READ	Tape = aab	Store = NULL
State = 3	READ	Tape = ab	Store = NULL
State = 4	READ	Tape = b	Store = NULL
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: baaba

State = 1	START	Tape = baaba	Store = NULL
State = 2	READ	Tape = aaba	Store = NULL
State = 2	READ	Tape = aba	Store = NULL
State = 3	READ	Tape = ba	Store = NULL
State = 4	READ	Tape = a	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 3	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: baabb

State = 1	START	Tape = baabb	Store = NULL
State = 2	READ	Tape = aabb	Store = NULL
State = 2	READ	Tape = abb	Store = NULL
State = 3	READ	Tape = bb	Store = NULL
State = 4	READ	Tape = b	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: babaa

State = 1	START	Tape = babaa	Store = NULL
State = 2	READ	Tape = abaa	Store = NULL
State = 2	READ	Tape = baa	Store = NULL
State = 3	READ	Tape = aa	Store = NULL
State = 2	READ	Tape = a	Store = NULL
State = 3	READ	Tape = NULL	Store = NULL
State = 4	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: babab

State = 1	START	Tape = babab	Store = NULL
State = 2	READ	Tape = abab	Store = NULL

```

State = 2 READ      Tape = bab      Store = NULL
State = 3 READ      Tape = ab       Store = NULL
State = 2 READ      Tape = b        Store = NULL
State = 3 READ      Tape = NULL     Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: babba

```

State = 1 START     Tape = babba    Store = NULL
State = 2 READ      Tape = abba     Store = NULL
State = 2 READ      Tape = bba      Store = NULL
State = 3 READ      Tape = ba       Store = NULL
State = 2 READ      Tape = a        Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 3 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: babbb

```

State = 1 START     Tape = babbb    Store = NULL
State = 2 READ      Tape = abbb     Store = NULL
State = 2 READ      Tape = bbb      Store = NULL
State = 3 READ      Tape = bb       Store = NULL
State = 2 READ      Tape = b        Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbaaa

```

State = 1 START     Tape = bbaaa    Store = NULL
State = 2 READ      Tape = baaa     Store = NULL
State = 2 READ      Tape = aaa      Store = NULL
State = 2 READ      Tape = aa       Store = NULL
State = 3 READ      Tape = a        Store = NULL
State = 4 READ      Tape = NULL     Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbaab

```

State = 1 START     Tape = bbaab    Store = NULL
State = 2 READ      Tape = baab     Store = NULL
State = 2 READ      Tape = aab      Store = NULL
State = 2 READ      Tape = ab       Store = NULL
State = 3 READ      Tape = b        Store = NULL
State = 4 READ      Tape = NULL     Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbaba

```

State = 1 START     Tape = bbaba    Store = NULL
State = 2 READ      Tape = baba     Store = NULL
State = 2 READ      Tape = aba      Store = NULL
State = 2 READ      Tape = ba       Store = NULL
State = 3 READ      Tape = a        Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 3 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbabb

```

State = 1 START     Tape = bbabb    Store = NULL
State = 2 READ      Tape = babb     Store = NULL

```

```

State = 2 READ      Tape = abb      Store = NULL
State = 2 READ      Tape = bb       Store = NULL
State = 3 READ      Tape = b        Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbbaa

```

State = 1 START     Tape = bbbaa    Store = NULL
State = 2 READ      Tape = bbaa     Store = NULL
State = 2 READ      Tape = baa      Store = NULL
State = 2 READ      Tape = aa       Store = NULL
State = 2 READ      Tape = a        Store = NULL
State = 3 READ      Tape = NULL     Store = NULL
State = 4 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbbab

```

State = 1 START     Tape = bbbab    Store = NULL
State = 2 READ      Tape = bbab     Store = NULL
State = 2 READ      Tape = bab      Store = NULL
State = 2 READ      Tape = ab       Store = NULL
State = 2 READ      Tape = b        Store = NULL
State = 3 READ      Tape = NULL     Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbbba

```

State = 1 START     Tape = bbbba    Store = NULL
State = 2 READ      Tape = bbba     Store = NULL
State = 2 READ      Tape = bba      Store = NULL
State = 2 READ      Tape = ba       Store = NULL
State = 2 READ      Tape = a        Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 3 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbbbb

```

State = 1 START     Tape = bbbbb    Store = NULL
State = 2 READ      Tape = bbbb     Store = NULL
State = 2 READ      Tape = bbb      Store = NULL
State = 2 READ      Tape = bb       Store = NULL
State = 2 READ      Tape = b        Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaaaa

```

State = 1 START     Tape = aaaaaa   Store = NULL
State = 2 READ      Tape = aaaaa    Store = NULL
State = 3 READ      Tape = aaaa     Store = NULL
State = 4 READ      Tape = aaa      Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaaab

```

State = 1 START     Tape = aaaaab   Store = NULL
State = 2 READ      Tape = aaaab    Store = NULL
State = 3 READ      Tape = aaab     Store = NULL
State = 4 READ      Tape = aab      Store = NULL

```

```

State = 0 REJECT

Enter word or 'N' for null string or '*' to turn off machine: aaaaba

State = 1 START      Tape = aaaaba      Store = NULL
State = 2 READ       Tape = aaaba       Store = NULL
State = 3 READ       Tape = aaba        Store = NULL
State = 4 READ       Tape = aba         Store = NULL
State = 0 REJECT

Enter word or 'N' for null string or '*' to turn off machine: aaaabb

State = 1 START      Tape = aaaabb      Store = NULL
State = 2 READ       Tape = aaabb       Store = NULL
State = 3 READ       Tape = aabb        Store = NULL
State = 4 READ       Tape = abb         Store = NULL
State = 0 REJECT

Enter word or 'N' for null string or '*' to turn off machine: aaabaa

State = 1 START      Tape = aaabaa      Store = NULL
State = 2 READ       Tape = aabaa       Store = NULL
State = 3 READ       Tape = abaa        Store = NULL
State = 4 READ       Tape = baa         Store = NULL
State = 0 REJECT

Enter word or 'N' for null string or '*' to turn off machine: aaabab

State = 1 START      Tape = aaabab      Store = NULL
State = 2 READ       Tape = aabab       Store = NULL
State = 3 READ       Tape = abab        Store = NULL
State = 4 READ       Tape = bab         Store = NULL
State = 0 REJECT

Enter word or 'N' for null string or '*' to turn off machine: aaabba

State = 1 START      Tape = aaabba      Store = NULL
State = 2 READ       Tape = aabba       Store = NULL
State = 3 READ       Tape = abba        Store = NULL
State = 4 READ       Tape = bba         Store = NULL
State = 0 REJECT

Enter word or 'N' for null string or '*' to turn off machine: aaabbb

State = 1 START      Tape = aaabbb      Store = NULL
State = 2 READ       Tape = aabbb       Store = NULL
State = 3 READ       Tape = abbb        Store = NULL
State = 4 READ       Tape = bbb         Store = NULL
State = 0 REJECT

Enter word or 'N' for null string or '*' to turn off machine: aabaaa

State = 1 START      Tape = aabaaa      Store = NULL
State = 2 READ       Tape = abaaa       Store = NULL
State = 3 READ       Tape = baaa        Store = NULL
State = 4 READ       Tape = aaa         Store = NULL
State = 2 READ       Tape = aa          Store = NULL
State = 3 READ       Tape = a           Store = NULL
State = 4 READ       Tape = NULL        Store = NULL
State = 0 REJECT

Enter word or 'N' for null string or '*' to turn off machine: aabaab

State = 1 START      Tape = aabaab      Store = NULL
State = 2 READ       Tape = abaab       Store = NULL

```

```

State = 3 READ      Tape = baab      Store = NULL
State = 4 READ      Tape = aab       Store = NULL
State = 2 READ      Tape = ab        Store = NULL
State = 3 READ      Tape = b         Store = NULL
State = 4 READ      Tape = NULL      Store = NULL
State = 2 READ      Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aababa

```

State = 1 START     Tape = aababa     Store = NULL
State = 2 READ      Tape = ababa      Store = NULL
State = 3 READ      Tape = baba       Store = NULL
State = 4 READ      Tape = aba        Store = NULL
State = 2 READ      Tape = ba         Store = NULL
State = 3 READ      Tape = a          Store = NULL
State = 2 READ      Tape = NULL       Store = NULL
State = 3 READ      Tape = NULL       Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aababb

```

State = 1 START     Tape = aababb     Store = NULL
State = 2 READ      Tape = ababb      Store = NULL
State = 3 READ      Tape = babb       Store = NULL
State = 4 READ      Tape = abb        Store = NULL
State = 2 READ      Tape = bb         Store = NULL
State = 3 READ      Tape = b          Store = NULL
State = 2 READ      Tape = NULL       Store = NULL
State = 2 READ      Tape = NULL       Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aabbaa

```

State = 1 START     Tape = aabbaa     Store = NULL
State = 2 READ      Tape = abbaa      Store = NULL
State = 3 READ      Tape = bbba       Store = NULL
State = 4 READ      Tape = baa        Store = NULL
State = 2 READ      Tape = aa         Store = NULL
State = 2 READ      Tape = a          Store = NULL
State = 3 READ      Tape = NULL       Store = NULL
State = 4 READ      Tape = NULL       Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aabbab

```

State = 1 START     Tape = aabbab     Store = NULL
State = 2 READ      Tape = abbab      Store = NULL
State = 3 READ      Tape = bbab       Store = NULL
State = 4 READ      Tape = bab        Store = NULL
State = 2 READ      Tape = ab         Store = NULL
State = 2 READ      Tape = b          Store = NULL
State = 3 READ      Tape = NULL       Store = NULL
State = 2 READ      Tape = NULL       Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aabbba

```

State = 1 START     Tape = aabbba     Store = NULL
State = 2 READ      Tape = abbba      Store = NULL
State = 3 READ      Tape = bbba       Store = NULL
State = 4 READ      Tape = bba        Store = NULL
State = 2 READ      Tape = ba         Store = NULL
State = 2 READ      Tape = a          Store = NULL
State = 2 READ      Tape = NULL       Store = NULL

```

```

State = 3 READ      Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aabbbb

```

State = 1 START      Tape = aabbbb      Store = NULL
State = 2 READ      Tape = abbbb      Store = NULL
State = 3 READ      Tape = bbbb      Store = NULL
State = 4 READ      Tape = bbb      Store = NULL
State = 2 READ      Tape = bb      Store = NULL
State = 2 READ      Tape = b      Store = NULL
State = 2 READ      Tape = NULL      Store = NULL
State = 2 READ      Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: abaaaa

```

State = 1 START      Tape = abaaaa      Store = NULL
State = 2 READ      Tape = baaaa      Store = NULL
State = 3 READ      Tape = aaaa      Store = NULL
State = 2 READ      Tape = aaa      Store = NULL
State = 3 READ      Tape = aa      Store = NULL
State = 4 READ      Tape = a      Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: abaaab

```

State = 1 START      Tape = abaaab      Store = NULL
State = 2 READ      Tape = baaab      Store = NULL
State = 3 READ      Tape = aaab      Store = NULL
State = 2 READ      Tape = aab      Store = NULL
State = 3 READ      Tape = ab      Store = NULL
State = 4 READ      Tape = b      Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: abaaba

```

State = 1 START      Tape = abaaba      Store = NULL
State = 2 READ      Tape = baaba      Store = NULL
State = 3 READ      Tape = aaba      Store = NULL
State = 2 READ      Tape = aba      Store = NULL
State = 3 READ      Tape = ba      Store = NULL
State = 4 READ      Tape = a      Store = NULL
State = 2 READ      Tape = NULL      Store = NULL
State = 3 READ      Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: abaabb

```

State = 1 START      Tape = abaabb      Store = NULL
State = 2 READ      Tape = baabb      Store = NULL
State = 3 READ      Tape = aabb      Store = NULL
State = 2 READ      Tape = abb      Store = NULL
State = 3 READ      Tape = bb      Store = NULL
State = 4 READ      Tape = b      Store = NULL
State = 2 READ      Tape = NULL      Store = NULL
State = 2 READ      Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: ababaa

```

State = 1 START      Tape = ababaa      Store = NULL
State = 2 READ      Tape = babaa      Store = NULL
State = 3 READ      Tape = abaa      Store = NULL
State = 2 READ      Tape = baa      Store = NULL

```

```

State = 3 READ      Tape = aa      Store = NULL
State = 2 READ      Tape = a       Store = NULL
State = 3 READ      Tape = NULL     Store = NULL
State = 4 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: ababab

```

State = 1 START     Tape = ababab   Store = NULL
State = 2 READ      Tape = babab    Store = NULL
State = 3 READ      Tape = abab     Store = NULL
State = 2 READ      Tape = bab      Store = NULL
State = 3 READ      Tape = ab       Store = NULL
State = 2 READ      Tape = b        Store = NULL
State = 3 READ      Tape = NULL     Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: ababba

```

State = 1 START     Tape = ababba   Store = NULL
State = 2 READ      Tape = babba    Store = NULL
State = 3 READ      Tape = abba     Store = NULL
State = 2 READ      Tape = bba      Store = NULL
State = 3 READ      Tape = ba       Store = NULL
State = 2 READ      Tape = a        Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 3 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: ababbb

```

State = 1 START     Tape = ababbb   Store = NULL
State = 2 READ      Tape = babbb    Store = NULL
State = 3 READ      Tape = abbb     Store = NULL
State = 2 READ      Tape = bbb      Store = NULL
State = 3 READ      Tape = bb       Store = NULL
State = 2 READ      Tape = b        Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: abbaaa

```

State = 1 START     Tape = abbaaa   Store = NULL
State = 2 READ      Tape = bbaaa    Store = NULL
State = 3 READ      Tape = baaa     Store = NULL
State = 2 READ      Tape = aaa      Store = NULL
State = 2 READ      Tape = aa       Store = NULL
State = 3 READ      Tape = a        Store = NULL
State = 4 READ      Tape = NULL     Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: abbaab

```

State = 1 START     Tape = abbaab   Store = NULL
State = 2 READ      Tape = bbaab    Store = NULL
State = 3 READ      Tape = baab     Store = NULL
State = 2 READ      Tape = aab      Store = NULL
State = 2 READ      Tape = ab       Store = NULL
State = 3 READ      Tape = b        Store = NULL
State = 4 READ      Tape = NULL     Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: abbaba

State =	1	START	Tape =	abbaba	Store =	NULL
State =	2	READ	Tape =	bbaba	Store =	NULL
State =	3	READ	Tape =	baba	Store =	NULL
State =	2	READ	Tape =	aba	Store =	NULL
State =	2	READ	Tape =	ba	Store =	NULL
State =	3	READ	Tape =	a	Store =	NULL
State =	2	READ	Tape =	NULL	Store =	NULL
State =	3	READ	Tape =	NULL	Store =	NULL
State =	5	ACCEPT				

Enter word or 'N' for null string or '\*' to turn off machine: abbabb

State =	1	START	Tape =	abbabb	Store =	NULL
State =	2	READ	Tape =	bbabb	Store =	NULL
State =	3	READ	Tape =	babb	Store =	NULL
State =	2	READ	Tape =	abb	Store =	NULL
State =	2	READ	Tape =	bb	Store =	NULL
State =	3	READ	Tape =	b	Store =	NULL
State =	2	READ	Tape =	NULL	Store =	NULL
State =	2	READ	Tape =	NULL	Store =	NULL
State =	5	ACCEPT				

Enter word or 'N' for null string or '\*' to turn off machine: abbbaa

State =	1	START	Tape =	abbbaa	Store =	NULL
State =	2	READ	Tape =	bbbaa	Store =	NULL
State =	3	READ	Tape =	bbaa	Store =	NULL
State =	2	READ	Tape =	baa	Store =	NULL
State =	2	READ	Tape =	aa	Store =	NULL
State =	2	READ	Tape =	a	Store =	NULL
State =	3	READ	Tape =	NULL	Store =	NULL
State =	4	READ	Tape =	NULL	Store =	NULL
State =	5	ACCEPT				

Enter word or 'N' for null string or '\*' to turn off machine: abbbab

State =	1	START	Tape =	abbbab	Store =	NULL
State =	2	READ	Tape =	bbbab	Store =	NULL
State =	3	READ	Tape =	bbab	Store =	NULL
State =	2	READ	Tape =	bab	Store =	NULL
State =	2	READ	Tape =	ab	Store =	NULL
State =	2	READ	Tape =	b	Store =	NULL
State =	3	READ	Tape =	NULL	Store =	NULL
State =	2	READ	Tape =	NULL	Store =	NULL
State =	5	ACCEPT				

Enter word or 'N' for null string or '\*' to turn off machine: abbbba

State =	1	START	Tape =	abbbba	Store =	NULL
State =	2	READ	Tape =	bbbba	Store =	NULL
State =	3	READ	Tape =	bbba	Store =	NULL
State =	2	READ	Tape =	bba	Store =	NULL
State =	2	READ	Tape =	ba	Store =	NULL
State =	2	READ	Tape =	a	Store =	NULL
State =	2	READ	Tape =	NULL	Store =	NULL
State =	3	READ	Tape =	NULL	Store =	NULL
State =	5	ACCEPT				

Enter word or 'N' for null string or '\*' to turn off machine: abbbbb

State =	1	START	Tape =	abbbbb	Store =	NULL
State =	2	READ	Tape =	bbbbb	Store =	NULL
State =	3	READ	Tape =	bbbb	Store =	NULL



```

State = 2 READ      Tape = bbb      Store = NULL
State = 2 READ      Tape = bb       Store = NULL
State = 2 READ      Tape = b        Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: baaaaa

```

State = 1 START     Tape = baaaaa   Store = NULL
State = 2 READ      Tape = aaaaa    Store = NULL
State = 2 READ      Tape = aaaa     Store = NULL
State = 3 READ      Tape = aaa      Store = NULL
State = 4 READ      Tape = aa       Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: baaaab

```

State = 1 START     Tape = baaaab   Store = NULL
State = 2 READ      Tape = aaaab    Store = NULL
State = 2 READ      Tape = aaab     Store = NULL
State = 3 READ      Tape = aab      Store = NULL
State = 4 READ      Tape = ab       Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: baaaba

```

State = 1 START     Tape = baaaba   Store = NULL
State = 2 READ      Tape = aaaba    Store = NULL
State = 2 READ      Tape = aaba     Store = NULL
State = 3 READ      Tape = aba      Store = NULL
State = 4 READ      Tape = ba       Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: baaabb

```

State = 1 START     Tape = baaabb   Store = NULL
State = 2 READ      Tape = aaabb    Store = NULL
State = 2 READ      Tape = aabb     Store = NULL
State = 3 READ      Tape = abb      Store = NULL
State = 4 READ      Tape = bb       Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: baabaa

```

State = 1 START     Tape = baabaa   Store = NULL
State = 2 READ      Tape = aabaa    Store = NULL
State = 2 READ      Tape = abaa     Store = NULL
State = 3 READ      Tape = baa      Store = NULL
State = 4 READ      Tape = aa       Store = NULL
State = 2 READ      Tape = a        Store = NULL
State = 3 READ      Tape = NULL     Store = NULL
State = 4 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: baabab

```

State = 1 START     Tape = baabab   Store = NULL
State = 2 READ      Tape = aabab    Store = NULL
State = 2 READ      Tape = abab     Store = NULL
State = 3 READ      Tape = bab      Store = NULL
State = 4 READ      Tape = ab       Store = NULL
State = 2 READ      Tape = b        Store = NULL
State = 3 READ      Tape = NULL     Store = NULL
State = 2 READ      Tape = NULL     Store = NULL

```

State = 5 ACCEPT

Enter word or 'N' for null string or '\*' to turn off machine: baabba

State = 1	START	Tape = baabba	Store = NULL
State = 2	READ	Tape = aabba	Store = NULL
State = 2	READ	Tape = abba	Store = NULL
State = 3	READ	Tape = bba	Store = NULL
State = 4	READ	Tape = ba	Store = NULL
State = 2	READ	Tape = a	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 3	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: baabbb

State = 1	START	Tape = baabbb	Store = NULL
State = 2	READ	Tape = aabbb	Store = NULL
State = 2	READ	Tape = abbb	Store = NULL
State = 3	READ	Tape = bbb	Store = NULL
State = 4	READ	Tape = bb	Store = NULL
State = 2	READ	Tape = b	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: babaaa

State = 1	START	Tape = babaaa	Store = NULL
State = 2	READ	Tape = abaaa	Store = NULL
State = 2	READ	Tape = baaa	Store = NULL
State = 3	READ	Tape = aaa	Store = NULL
State = 2	READ	Tape = aa	Store = NULL
State = 3	READ	Tape = a	Store = NULL
State = 4	READ	Tape = NULL	Store = NULL
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: babaab

State = 1	START	Tape = babaab	Store = NULL
State = 2	READ	Tape = abaab	Store = NULL
State = 2	READ	Tape = baab	Store = NULL
State = 3	READ	Tape = aab	Store = NULL
State = 2	READ	Tape = ab	Store = NULL
State = 3	READ	Tape = b	Store = NULL
State = 4	READ	Tape = NULL	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL

Enter word or 'N' for null string or '\*' to turn off machine: bababa

State = 1	START	Tape = bababa	Store = NULL
State = 2	READ	Tape = ababa	Store = NULL
State = 2	READ	Tape = baba	Store = NULL
State = 3	READ	Tape = aba	Store = NULL
State = 2	READ	Tape = ba	Store = NULL
State = 3	READ	Tape = a	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 3	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: bababb

State = 1	START	Tape = bababb	Store = NULL
State = 2	READ	Tape = ababb	Store = NULL
State = 2	READ	Tape = babb	Store = NULL

```

State = 3 READ      Tape = abb      Store = NULL
State = 2 READ      Tape = bb       Store = NULL
State = 3 READ      Tape = b        Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT
Enter word or 'N' for null string or '*' to turn off machine: babbaa

```

```

State = 1 START     Tape = babbaa   Store = NULL
State = 2 READ      Tape = abbaa    Store = NULL
State = 2 READ      Tape = bbaa     Store = NULL
State = 3 READ      Tape = baa      Store = NULL
State = 2 READ      Tape = aa       Store = NULL
State = 2 READ      Tape = a        Store = NULL
State = 3 READ      Tape = NULL     Store = NULL
State = 4 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: babbab

```

State = 1 START     Tape = babbab   Store = NULL
State = 2 READ      Tape = abbab    Store = NULL
State = 2 READ      Tape = bbab     Store = NULL
State = 3 READ      Tape = bab      Store = NULL
State = 2 READ      Tape = ab       Store = NULL
State = 2 READ      Tape = b        Store = NULL
State = 3 READ      Tape = NULL     Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: babbbba

```

State = 1 START     Tape = babbbba  Store = NULL
State = 2 READ      Tape = abbbba   Store = NULL
State = 2 READ      Tape = bbba     Store = NULL
State = 3 READ      Tape = bba      Store = NULL
State = 2 READ      Tape = ba       Store = NULL
State = 2 READ      Tape = a        Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 3 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: babbbbb

```

State = 1 START     Tape = babbbbb  Store = NULL
State = 2 READ      Tape = abbbbb   Store = NULL
State = 2 READ      Tape = bbbb     Store = NULL
State = 3 READ      Tape = bbb      Store = NULL
State = 2 READ      Tape = bb       Store = NULL
State = 2 READ      Tape = b        Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 2 READ      Tape = NULL     Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbaaaaa

```

State = 1 START     Tape = bbaaaaa  Store = NULL
State = 2 READ      Tape = baaaaa   Store = NULL
State = 2 READ      Tape = aaaa     Store = NULL
State = 2 READ      Tape = aaa      Store = NULL
State = 3 READ      Tape = aa       Store = NULL
State = 4 READ      Tape = a        Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbaaab

State = 1	START	Tape = bbaaab	Store = NULL
State = 2	READ	Tape = baaab	Store = NULL
State = 2	READ	Tape = aaab	Store = NULL
State = 2	READ	Tape = aab	Store = NULL
State = 3	READ	Tape = ab	Store = NULL
State = 4	READ	Tape = b	Store = NULL
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: bbaaba

State = 1	START	Tape = bbaaba	Store = NULL
State = 2	READ	Tape = baaba	Store = NULL
State = 2	READ	Tape = aaba	Store = NULL
State = 2	READ	Tape = aba	Store = NULL
State = 3	READ	Tape = ba	Store = NULL
State = 4	READ	Tape = a	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 3	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: bbaabb

State = 1	START	Tape = bbaabb	Store = NULL
State = 2	READ	Tape = baabb	Store = NULL
State = 2	READ	Tape = aabb	Store = NULL
State = 2	READ	Tape = abb	Store = NULL
State = 3	READ	Tape = bb	Store = NULL
State = 4	READ	Tape = b	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: bbabaa

State = 1	START	Tape = bbabaa	Store = NULL
State = 2	READ	Tape = babaa	Store = NULL
State = 2	READ	Tape = abaa	Store = NULL
State = 2	READ	Tape = baa	Store = NULL
State = 3	READ	Tape = aa	Store = NULL
State = 2	READ	Tape = a	Store = NULL
State = 3	READ	Tape = NULL	Store = NULL
State = 4	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: bbabab

State = 1	START	Tape = bbabab	Store = NULL
State = 2	READ	Tape = babab	Store = NULL
State = 2	READ	Tape = abab	Store = NULL
State = 2	READ	Tape = bab	Store = NULL
State = 3	READ	Tape = ab	Store = NULL
State = 2	READ	Tape = b	Store = NULL
State = 3	READ	Tape = NULL	Store = NULL
State = 2	READ	Tape = NULL	Store = NULL
State = 5	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: bbabba

State = 1	START	Tape = bbabba	Store = NULL
State = 2	READ	Tape = babba	Store = NULL
State = 2	READ	Tape = abba	Store = NULL
State = 2	READ	Tape = bba	Store = NULL
State = 3	READ	Tape = ba	Store = NULL
State = 2	READ	Tape = a	Store = NULL

```

State = 2 READ      Tape = NULL      Store = NULL
State = 3 READ      Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbabbb

```

State = 1 START      Tape = bbabbb      Store = NULL
State = 2 READ      Tape = babbb      Store = NULL
State = 2 READ      Tape = abbb      Store = NULL
State = 2 READ      Tape = bbb      Store = NULL
State = 3 READ      Tape = bb      Store = NULL
State = 2 READ      Tape = b      Store = NULL
State = 2 READ      Tape = NULL      Store = NULL
State = 2 READ      Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbbaaa

```

State = 1 START      Tape = bbbaaa      Store = NULL
State = 2 READ      Tape = bbaaa      Store = NULL
State = 2 READ      Tape = baaa      Store = NULL
State = 2 READ      Tape = aaa      Store = NULL
State = 2 READ      Tape = aa      Store = NULL
State = 3 READ      Tape = a      Store = NULL
State = 4 READ      Tape = NULL      Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbbaab

```

State = 1 START      Tape = bbbaab      Store = NULL
State = 2 READ      Tape = bbaab      Store = NULL
State = 2 READ      Tape = baab      Store = NULL
State = 2 READ      Tape = aab      Store = NULL
State = 2 READ      Tape = ab      Store = NULL
State = 3 READ      Tape = b      Store = NULL
State = 4 READ      Tape = NULL      Store = NULL
State = 2 READ      Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbbaba

```

State = 1 START      Tape = bbbaba      Store = NULL
State = 2 READ      Tape = bbaba      Store = NULL
State = 2 READ      Tape = baba      Store = NULL
State = 2 READ      Tape = aba      Store = NULL
State = 2 READ      Tape = ba      Store = NULL
State = 3 READ      Tape = a      Store = NULL
State = 2 READ      Tape = NULL      Store = NULL
State = 3 READ      Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbbabb

```

State = 1 START      Tape = bbbabb      Store = NULL
State = 2 READ      Tape = bbabb      Store = NULL
State = 2 READ      Tape = babb      Store = NULL
State = 2 READ      Tape = abb      Store = NULL
State = 2 READ      Tape = bb      Store = NULL
State = 3 READ      Tape = b      Store = NULL
State = 2 READ      Tape = NULL      Store = NULL
State = 2 READ      Tape = NULL      Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbbbaa

```

State = 1 START      Tape = bbbbaa      Store = NULL
State = 2 READ       Tape = bbbbaa      Store = NULL
State = 2 READ       Tape = bbbaa       Store = NULL
State = 2 READ       Tape = baa         Store = NULL
State = 2 READ       Tape = aa          Store = NULL
State = 2 READ       Tape = a           Store = NULL
State = 3 READ       Tape = NULL        Store = NULL
State = 4 READ       Tape = NULL        Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbbbab

```

State = 1 START      Tape = bbbbab      Store = NULL
State = 2 READ       Tape = bbbbab      Store = NULL
State = 2 READ       Tape = bbab        Store = NULL
State = 2 READ       Tape = bab         Store = NULL
State = 2 READ       Tape = ab          Store = NULL
State = 2 READ       Tape = b           Store = NULL
State = 3 READ       Tape = NULL        Store = NULL
State = 2 READ       Tape = NULL        Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbbbbba

```

State = 1 START      Tape = bbbbbba     Store = NULL
State = 2 READ       Tape = bbbbaa      Store = NULL
State = 2 READ       Tape = bbba        Store = NULL
State = 2 READ       Tape = bba         Store = NULL
State = 2 READ       Tape = ba          Store = NULL
State = 2 READ       Tape = a           Store = NULL
State = 2 READ       Tape = NULL        Store = NULL
State = 3 READ       Tape = NULL        Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbbbbb

```

State = 1 START      Tape = bbbbbb      Store = NULL
State = 2 READ       Tape = bbbbbb      Store = NULL
State = 2 READ       Tape = bbbb        Store = NULL
State = 2 READ       Tape = bbb         Store = NULL
State = 2 READ       Tape = bb          Store = NULL
State = 2 READ       Tape = b           Store = NULL
State = 2 READ       Tape = NULL        Store = NULL
State = 2 READ       Tape = NULL        Store = NULL
State = 5 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaaaaa

```

State = 1 START      Tape = aaaaaaa     Store = NULL
State = 2 READ       Tape = aaaaaa      Store = NULL
State = 3 READ       Tape = aaaaa       Store = NULL
State = 4 READ       Tape = aaaa        Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaaaab

```

State = 1 START      Tape = aaaaaab     Store = NULL
State = 2 READ       Tape = aaaaaab     Store = NULL
State = 3 READ       Tape = aaaab       Store = NULL
State = 4 READ       Tape = aaab        Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaaaba

```

State = 1 START      Tape = aaaaaba     Store = NULL

```

```

State = 2 READ      Tape = aaaaba      Store = NULL
State = 3 READ      Tape = aaaba       Store = NULL
State = 4 READ      Tape = aaba        Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaaabb

```

State = 1 START     Tape = aaaaabb     Store = NULL
State = 2 READ      Tape = aaaabb      Store = NULL
State = 3 READ      Tape = aaabb       Store = NULL
State = 4 READ      Tape = aabb        Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaabaa

```

State = 1 START     Tape = aaaabaa     Store = NULL
State = 2 READ      Tape = aaabaa      Store = NULL
State = 3 READ      Tape = aabaa       Store = NULL
State = 4 READ      Tape = abaa        Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaabab

```

State = 1 START     Tape = aaaabab     Store = NULL
State = 2 READ      Tape = aaabab      Store = NULL
State = 3 READ      Tape = aabab       Store = NULL
State = 4 READ      Tape = abab        Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaabba

```

State = 1 START     Tape = aaaabba     Store = NULL
State = 2 READ      Tape = aaabba      Store = NULL
State = 3 READ      Tape = aabba       Store = NULL
State = 4 READ      Tape = abba        Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaabbb

```

State = 1 START     Tape = aaaabbb     Store = NULL
State = 2 READ      Tape = aaabbb      Store = NULL
State = 3 READ      Tape = aabbb       Store = NULL
State = 4 READ      Tape = abbb        Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaabaaa

```

State = 1 START     Tape = aaabaaa     Store = NULL
State = 2 READ      Tape = aabaaa      Store = NULL
State = 3 READ      Tape = abaaa       Store = NULL
State = 4 READ      Tape = baaa        Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaabaab

```

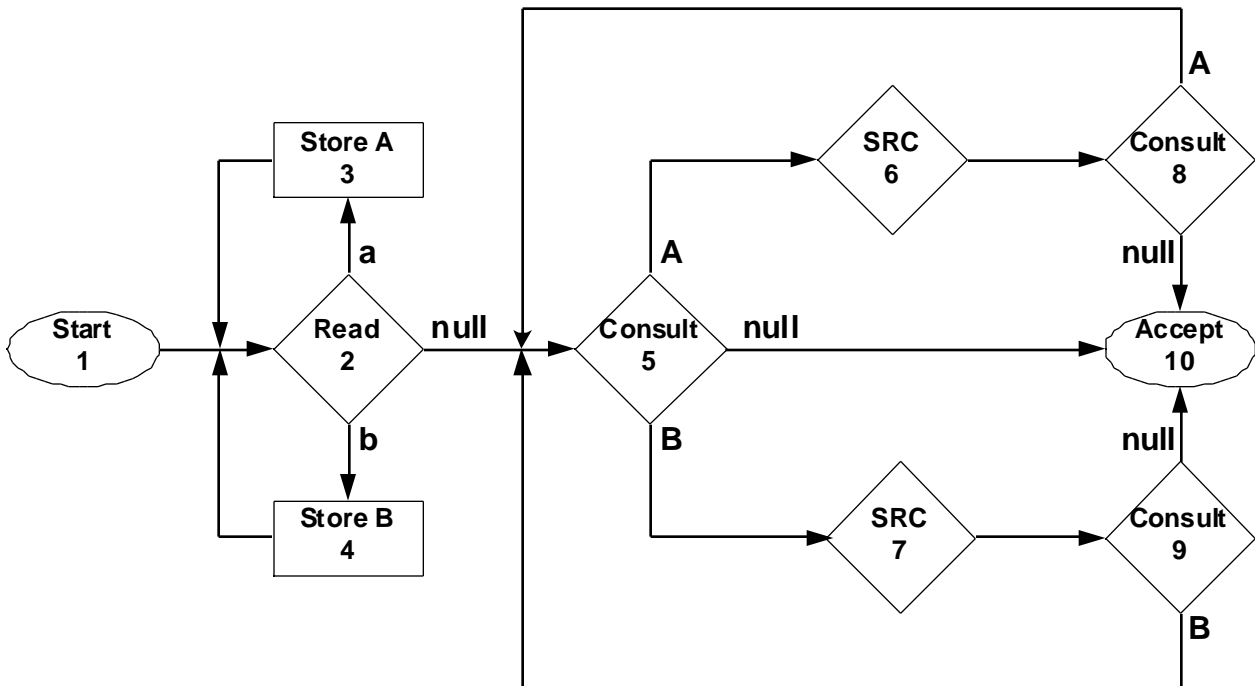
State = 1 START     Tape = aaabaab     Store = NULL
State = 2 READ      Tape = aabaab      Store = NULL
State = 3 READ      Tape = abaab       Store = NULL
State = 4 READ      Tape = baab        Store = NULL
State = 0 REJECT

```

## -----Palindromes-----

**Table 5. Post Machine describing the language Palindromes**

Number	State Type	Add Letter	Next State	Read/Branch			Consult/Branch		
				If a	If b	If null	If A	If B	If null
0	Reject								
1	Start		2						
2	Read			3	4	5			
3	Store	A	2						
4	Store	B	2						
5	Consult						6	7	10
6	SRC		8						
7	SRC		9				0	0	8
8	Consult						5	0	10
9	Consult						0	5	10
10	Accept								



Menu:

1: Run Post Machine  
2: Quit

Enter: 1

Enter Tape alphabet:

Enter the size of the alphabet: 2

Enter letter: a

Enter letter: b



```

Alphabet:
Letter 1: a
Letter 2: b

Enter Store alphabet:

Enter the size of the alphabet: 2
Enter letter: A
Enter letter: B

Alphabet:
Letter 1: A
Letter 2: B

Defining Post Machine:

Enter the number of states excluding rejected states: 10

State 1: START

Enter the next state (integer) : 2

State 2:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 1

Enter Next State (integer) :
    If a: 3
    If b: 4
    If NULL: 5

State 3:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 3
Enter letter to add at end of queue: A

Enter the next state (integer) : 2

State 4:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

```

Enter a number: 3  
Enter letter to add at end of queue: B

Enter the next state (integer) : 2

State 5:

State Type:  
1: Read Tape  
2: Consult Queue  
3: Store Queue  
4: Shift Right Cyclically (SRC)  
5: Add Front (AF)  
6: Accept State

Enter a number: 2

Enter Next State (integer) :  
    If A: 6  
    If B: 7  
    If NULL: 10

State 6:

State Type:  
1: Read Tape  
2: Consult Queue  
3: Store Queue  
4: Shift Right Cyclically (SRC)  
5: Add Front (AF)  
6: Accept State

Enter a number: 4

Enter the next state (integer) : 8

State 7:

State Type:  
1: Read Tape  
2: Consult Queue  
3: Store Queue  
4: Shift Right Cyclically (SRC)  
5: Add Front (AF)  
6: Accept State

Enter a number: 4

Enter the next state (integer) : 9

State 8:

State Type:  
1: Read Tape  
2: Consult Queue  
3: Store Queue  
4: Shift Right Cyclically (SRC)  
5: Add Front (AF)  
6: Accept State

Enter a number: 2

Enter Next State (integer) :  
    If A: 5  
    If B: 0

```

    If NULL: 10

State 9:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 2

Enter Next State (integer) :
    If A: 0
    If B: 5
    If NULL: 10

State 10:

State Type:
1: Read Tape
2: Consult Queue
3: Store Queue
4: Shift Right Cyclically (SRC)
5: Add Front (AF)
6: Accept State

Enter a number: 6

Print Transition Table:

Number State Add Queue Next If a If b If NULL If A If B If NULL
0       X      -      0    0    0    0    0    0    0
1       I      -      2    0    0    0    0    0    0
2       R      -      0    3    4    5    0    0    0
3       W      A      2    0    0    0    0    0    0
4       W      B      2    0    0    0    0    0    0
5       K      -      0    0    0    0    6    7    10
6       S      -      8    0    0    0    0    0    0
7       S      -      9    0    0    0    0    0    0
8       K      -      0    0    0    0    5    0    10
9       K      -      0    0    0    0    0    5    10
10      A      -      0    0    0    0    0    0    0

Legend for State Type:

I = Start,      A = Accept,      X=Reject
R = Read,      K = Consult,      W = Store,
S=Shift Right Cyclically,      F = Add Front

State = 1 START      Tape = NULL      Store = NULL
State = 2 READ       Tape = NULL      Store = NULL
State = 5 CONSULT    Tape = NULL      Store = NULL
State = 10 ACCEPT

Enter word or 'N' for null string or '*' to turn off machine: a

State = 1 START      Tape = a      Store = NULL
State = 2 READ       Tape = NULL      Store = NULL
State = 3 STORE      Tape = NULL      Store = A
State = 2 READ       Tape = NULL      Store = A
State = 5 CONSULT    Tape = NULL      Store = NULL

```

```

State = 6 SRC          Tape = NULL      Store = NULL
State = 8 CONSULT      Tape = NULL      Store = NULL
State = 10 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: b

```

State = 1 START        Tape = b         Store = NULL
State = 2 READ          Tape = NULL      Store = NULL
State = 4 STORE          Tape = NULL      Store = B
State = 2 READ          Tape = NULL      Store = B
State = 5 CONSULT       Tape = NULL      Store = NULL
State = 7 SRC           Tape = NULL      Store = NULL
State = 9 CONSULT       Tape = NULL      Store = NULL
State = 10 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aa

```

State = 1 START        Tape = aa        Store = NULL
State = 2 READ          Tape = a         Store = NULL
State = 3 STORE          Tape = a         Store = A
State = 2 READ          Tape = NULL      Store = A
State = 3 STORE          Tape = NULL      Store = AA
State = 2 READ          Tape = NULL      Store = AA
State = 5 CONSULT       Tape = NULL      Store = A
State = 6 SRC           Tape = NULL      Store = A
State = 8 CONSULT       Tape = NULL      Store = NULL
State = 5 CONSULT       Tape = NULL      Store = NULL
State = 10 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: ab

```

State = 1 START        Tape = ab        Store = NULL
State = 2 READ          Tape = b         Store = NULL
State = 3 STORE          Tape = b         Store = A
State = 2 READ          Tape = NULL      Store = A
State = 4 STORE          Tape = NULL      Store = AB
State = 2 READ          Tape = NULL      Store = AB
State = 5 CONSULT       Tape = NULL      Store = B
State = 6 SRC           Tape = NULL      Store = B
State = 8 CONSULT       Tape = NULL      Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: ba

```

State = 1 START        Tape = ba        Store = NULL
State = 2 READ          Tape = a         Store = NULL
State = 4 STORE          Tape = a         Store = B
State = 2 READ          Tape = NULL      Store = B
State = 3 STORE          Tape = NULL      Store = BA
State = 2 READ          Tape = NULL      Store = BA
State = 5 CONSULT       Tape = NULL      Store = A
State = 7 SRC           Tape = NULL      Store = A
State = 9 CONSULT       Tape = NULL      Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: bb

```

State = 1 START        Tape = bb        Store = NULL
State = 2 READ          Tape = b         Store = NULL
State = 4 STORE          Tape = b         Store = B
State = 2 READ          Tape = NULL      Store = B
State = 4 STORE          Tape = NULL      Store = BB
State = 2 READ          Tape = NULL      Store = BB
State = 5 CONSULT       Tape = NULL      Store = B
State = 7 SRC           Tape = NULL      Store = B

```

```

State = 9 CONSULT    Tape = NULL    Store = NULL
State = 5 CONSULT    Tape = NULL    Store = NULL
State = 10 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaa

```

State = 1 START      Tape = aaa      Store = NULL
State = 2 READ       Tape = aa       Store = NULL
State = 3 STORE      Tape = aa       Store = A
State = 2 READ       Tape = a        Store = A
State = 3 STORE      Tape = a        Store = AA
State = 2 READ       Tape = NULL     Store = AA
State = 3 STORE      Tape = NULL     Store = AAA
State = 2 READ       Tape = NULL     Store = AAA
State = 5 CONSULT    Tape = NULL     Store = AA
State = 6 SRC        Tape = NULL     Store = AA
State = 8 CONSULT    Tape = NULL     Store = A
State = 5 CONSULT    Tape = NULL     Store = NULL
State = 6 SRC        Tape = NULL     Store = NULL
State = 8 CONSULT    Tape = NULL     Store = NULL
State = 10 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aab

```

State = 1 START      Tape = aab      Store = NULL
State = 2 READ       Tape = ab       Store = NULL
State = 3 STORE      Tape = ab       Store = A
State = 2 READ       Tape = b        Store = A
State = 3 STORE      Tape = b        Store = AA
State = 2 READ       Tape = NULL     Store = AA
State = 4 STORE      Tape = NULL     Store = AAB
State = 2 READ       Tape = NULL     Store = AAB
State = 5 CONSULT    Tape = NULL     Store = AB
State = 6 SRC        Tape = NULL     Store = BA
State = 8 CONSULT    Tape = NULL     Store = A
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aba

```

State = 1 START      Tape = aba      Store = NULL
State = 2 READ       Tape = ba       Store = NULL
State = 3 STORE      Tape = ba       Store = A
State = 2 READ       Tape = a        Store = A
State = 4 STORE      Tape = a        Store = AB
State = 2 READ       Tape = NULL     Store = AB
State = 3 STORE      Tape = NULL     Store = ABA
State = 2 READ       Tape = NULL     Store = ABA
State = 5 CONSULT    Tape = NULL     Store = BA
State = 6 SRC        Tape = NULL     Store = AB
State = 8 CONSULT    Tape = NULL     Store = B
State = 5 CONSULT    Tape = NULL     Store = NULL
State = 7 SRC        Tape = NULL     Store = NULL
State = 9 CONSULT    Tape = NULL     Store = NULL
State = 10 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: abb

```

State = 1 START      Tape = abb      Store = NULL
State = 2 READ       Tape = bb       Store = NULL
State = 3 STORE      Tape = bb       Store = A
State = 2 READ       Tape = b        Store = A
State = 4 STORE      Tape = b        Store = AB
State = 2 READ       Tape = NULL     Store = AB
State = 4 STORE      Tape = NULL     Store = ABB
State = 2 READ       Tape = NULL     Store = ABB

```

```

State = 5 CONSULT    Tape = NULL    Store = BB
State = 6 SRC        Tape = NULL    Store = BB
State = 8 CONSULT    Tape = NULL    Store = B
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: baa

```

State = 1 START      Tape = baa     Store = NULL
State = 2 READ       Tape = aa      Store = NULL
State = 4 STORE      Tape = aa      Store = B
State = 2 READ       Tape = a       Store = B
State = 3 STORE      Tape = a       Store = BA
State = 2 READ       Tape = NULL     Store = BA
State = 3 STORE      Tape = NULL     Store = BAA
State = 2 READ       Tape = NULL     Store = BAA
State = 5 CONSULT    Tape = NULL     Store = AA
State = 7 SRC        Tape = NULL     Store = AA
State = 9 CONSULT    Tape = NULL     Store = A
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: bab

```

State = 1 START      Tape = bab     Store = NULL
State = 2 READ       Tape = ab      Store = NULL
State = 4 STORE      Tape = ab      Store = B
State = 2 READ       Tape = b       Store = B
State = 3 STORE      Tape = b       Store = BA
State = 2 READ       Tape = NULL     Store = BA
State = 4 STORE      Tape = NULL     Store = BAB
State = 2 READ       Tape = NULL     Store = BAB
State = 5 CONSULT    Tape = NULL     Store = AB
State = 7 SRC        Tape = NULL     Store = BA
State = 9 CONSULT    Tape = NULL     Store = A
State = 5 CONSULT    Tape = NULL     Store = NULL
State = 6 SRC        Tape = NULL     Store = NULL
State = 8 CONSULT    Tape = NULL     Store = NULL
State = 10 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: bba

```

State = 1 START      Tape = bba     Store = NULL
State = 2 READ       Tape = ba      Store = NULL
State = 4 STORE      Tape = ba      Store = B
State = 2 READ       Tape = a       Store = B
State = 4 STORE      Tape = a       Store = BB
State = 2 READ       Tape = NULL     Store = BB
State = 3 STORE      Tape = NULL     Store = BBA
State = 2 READ       Tape = NULL     Store = BBA
State = 5 CONSULT    Tape = NULL     Store = BA
State = 7 SRC        Tape = NULL     Store = AB
State = 9 CONSULT    Tape = NULL     Store = B
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: bbb

```

State = 1 START      Tape = bbb     Store = NULL
State = 2 READ       Tape = bb      Store = NULL
State = 4 STORE      Tape = bb      Store = B
State = 2 READ       Tape = b       Store = B
State = 4 STORE      Tape = b       Store = BB
State = 2 READ       Tape = NULL     Store = BB
State = 4 STORE      Tape = NULL     Store = BBB
State = 2 READ       Tape = NULL     Store = BBB
State = 5 CONSULT    Tape = NULL     Store = BB
State = 7 SRC        Tape = NULL     Store = BB

```

```

State = 9 CONSULT    Tape = NULL    Store = B
State = 5 CONSULT    Tape = NULL    Store = NULL
State = 7 SRC        Tape = NULL    Store = NULL
State = 9 CONSULT    Tape = NULL    Store = NULL
State = 10 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaaa

```

State = 1 START      Tape = aaaa    Store = NULL
State = 2 READ       Tape = aaa     Store = NULL
State = 3 STORE      Tape = aaa     Store = A
State = 2 READ       Tape = aa      Store = A
State = 3 STORE      Tape = aa      Store = AA
State = 2 READ       Tape = a       Store = AA
State = 3 STORE      Tape = a       Store = AAA
State = 2 READ       Tape = NULL    Store = AAA
State = 3 STORE      Tape = NULL    Store = AAAA
State = 2 READ       Tape = NULL    Store = AAAA
State = 5 CONSULT    Tape = NULL    Store = AAA
State = 6 SRC        Tape = NULL    Store = AAA
State = 8 CONSULT    Tape = NULL    Store = AA
State = 5 CONSULT    Tape = NULL    Store = A
State = 6 SRC        Tape = NULL    Store = A
State = 8 CONSULT    Tape = NULL    Store = NULL
State = 5 CONSULT    Tape = NULL    Store = NULL
State = 10 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaab

```

State = 1 START      Tape = aaab    Store = NULL
State = 2 READ       Tape = aab     Store = NULL
State = 3 STORE      Tape = aab     Store = A
State = 2 READ       Tape = ab      Store = A
State = 3 STORE      Tape = ab      Store = AA
State = 2 READ       Tape = b       Store = AA
State = 3 STORE      Tape = b       Store = AAA
State = 2 READ       Tape = NULL    Store = AAA
State = 4 STORE      Tape = NULL    Store = AAAB
State = 2 READ       Tape = NULL    Store = AAAB
State = 5 CONSULT    Tape = NULL    Store = AAB
State = 6 SRC        Tape = NULL    Store = BAA
State = 8 CONSULT    Tape = NULL    Store = AA
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aaba

```

State = 1 START      Tape = aaba    Store = NULL
State = 2 READ       Tape = aba     Store = NULL
State = 3 STORE      Tape = aba     Store = A
State = 2 READ       Tape = ba      Store = A
State = 3 STORE      Tape = ba      Store = AA
State = 2 READ       Tape = a       Store = AA
State = 4 STORE      Tape = a       Store = AAB
State = 2 READ       Tape = NULL    Store = AAB
State = 3 STORE      Tape = NULL    Store = AABA
State = 2 READ       Tape = NULL    Store = AABA
State = 5 CONSULT    Tape = NULL    Store = ABA
State = 6 SRC        Tape = NULL    Store = AAB
State = 8 CONSULT    Tape = NULL    Store = AB
State = 5 CONSULT    Tape = NULL    Store = B
State = 6 SRC        Tape = NULL    Store = B
State = 8 CONSULT    Tape = NULL    Store = NULL
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aabb

State =	1	START	Tape =	aabb	Store =	NULL
State =	2	READ	Tape =	abb	Store =	NULL
State =	3	STORE	Tape =	abb	Store =	A
State =	2	READ	Tape =	bb	Store =	A
State =	3	STORE	Tape =	bb	Store =	AA
State =	2	READ	Tape =	b	Store =	AA
State =	4	STORE	Tape =	b	Store =	AAB
State =	2	READ	Tape =	NULL	Store =	AAB
State =	4	STORE	Tape =	NULL	Store =	AABB
State =	2	READ	Tape =	NULL	Store =	AABB
State =	5	CONSULT	Tape =	NULL	Store =	ABB
State =	6	SRC	Tape =	NULL	Store =	BAB
State =	8	CONSULT	Tape =	NULL	Store =	AB
State =	0	REJECT				

Enter word or 'N' for null string or '\*' to turn off machine: abaa

State =	1	START	Tape =	abaa	Store =	NULL
State =	2	READ	Tape =	baa	Store =	NULL
State =	3	STORE	Tape =	baa	Store =	A
State =	2	READ	Tape =	aa	Store =	A
State =	4	STORE	Tape =	aa	Store =	AB
State =	2	READ	Tape =	a	Store =	AB
State =	3	STORE	Tape =	a	Store =	ABA
State =	2	READ	Tape =	NULL	Store =	ABA
State =	3	STORE	Tape =	NULL	Store =	ABAA
State =	2	READ	Tape =	NULL	Store =	ABAA
State =	5	CONSULT	Tape =	NULL	Store =	BAA
State =	6	SRC	Tape =	NULL	Store =	ABA
State =	8	CONSULT	Tape =	NULL	Store =	BA
State =	5	CONSULT	Tape =	NULL	Store =	A
State =	7	SRC	Tape =	NULL	Store =	A
State =	9	CONSULT	Tape =	NULL	Store =	NULL
State =	0	REJECT				

Enter word or 'N' for null string or '\*' to turn off machine: abab

State =	1	START	Tape =	abab	Store =	NULL
State =	2	READ	Tape =	bab	Store =	NULL
State =	3	STORE	Tape =	bab	Store =	A
State =	2	READ	Tape =	ab	Store =	A
State =	4	STORE	Tape =	ab	Store =	AB
State =	2	READ	Tape =	b	Store =	AB
State =	3	STORE	Tape =	b	Store =	ABA
State =	2	READ	Tape =	NULL	Store =	ABA
State =	4	STORE	Tape =	NULL	Store =	ABAB
State =	2	READ	Tape =	NULL	Store =	ABAB
State =	5	CONSULT	Tape =	NULL	Store =	BAB
State =	6	SRC	Tape =	NULL	Store =	BBA
State =	8	CONSULT	Tape =	NULL	Store =	BA
State =	0	REJECT				

Enter word or 'N' for null string or '\*' to turn off machine: abba

State =	1	START	Tape =	abba	Store =	NULL
State =	2	READ	Tape =	bba	Store =	NULL
State =	3	STORE	Tape =	bba	Store =	A
State =	2	READ	Tape =	ba	Store =	A
State =	4	STORE	Tape =	ba	Store =	AB
State =	2	READ	Tape =	a	Store =	AB
State =	4	STORE	Tape =	a	Store =	ABB
State =	2	READ	Tape =	NULL	Store =	ABB
State =	3	STORE	Tape =	NULL	Store =	ABBA
State =	2	READ	Tape =	NULL	Store =	ABBA



State = 5	CONSULT	Tape = NULL	Store = BBA
State = 6	SRC	Tape = NULL	Store = ABB
State = 8	CONSULT	Tape = NULL	Store = BB
State = 5	CONSULT	Tape = NULL	Store = B
State = 7	SRC	Tape = NULL	Store = B
State = 9	CONSULT	Tape = NULL	Store = NULL
State = 5	CONSULT	Tape = NULL	Store = NULL
State = 10	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: abbb

State = 1	START	Tape = abbb	Store = NULL
State = 2	READ	Tape = bbb	Store = NULL
State = 3	STORE	Tape = bbb	Store = A
State = 2	READ	Tape = bb	Store = A
State = 4	STORE	Tape = bb	Store = AB
State = 2	READ	Tape = b	Store = AB
State = 4	STORE	Tape = b	Store = ABB
State = 2	READ	Tape = NULL	Store = ABB
State = 4	STORE	Tape = NULL	Store = ABBB
State = 2	READ	Tape = NULL	Store = ABBB
State = 5	CONSULT	Tape = NULL	Store = BBB
State = 6	SRC	Tape = NULL	Store = BBB
State = 8	CONSULT	Tape = NULL	Store = BB
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: baaa

State = 1	START	Tape = baaa	Store = NULL
State = 2	READ	Tape = aaa	Store = NULL
State = 4	STORE	Tape = aaa	Store = B
State = 2	READ	Tape = aa	Store = B
State = 3	STORE	Tape = aa	Store = BA
State = 2	READ	Tape = a	Store = BA
State = 3	STORE	Tape = a	Store = BAA
State = 2	READ	Tape = NULL	Store = BAA
State = 3	STORE	Tape = NULL	Store = BAAA
State = 2	READ	Tape = NULL	Store = BAAA
State = 5	CONSULT	Tape = NULL	Store = AAA
State = 7	SRC	Tape = NULL	Store = AAA
State = 9	CONSULT	Tape = NULL	Store = AA
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: baab

State = 1	START	Tape = baab	Store = NULL
State = 2	READ	Tape = aab	Store = NULL
State = 4	STORE	Tape = aab	Store = B
State = 2	READ	Tape = ab	Store = B
State = 3	STORE	Tape = ab	Store = BA
State = 2	READ	Tape = b	Store = BA
State = 3	STORE	Tape = b	Store = BAA
State = 2	READ	Tape = NULL	Store = BAA
State = 4	STORE	Tape = NULL	Store = BAAB
State = 2	READ	Tape = NULL	Store = BAAB
State = 5	CONSULT	Tape = NULL	Store = AAB
State = 7	SRC	Tape = NULL	Store = BAA
State = 9	CONSULT	Tape = NULL	Store = AA
State = 5	CONSULT	Tape = NULL	Store = A
State = 6	SRC	Tape = NULL	Store = A
State = 8	CONSULT	Tape = NULL	Store = NULL
State = 5	CONSULT	Tape = NULL	Store = NULL
State = 10	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: baba

State = 1	START	Tape = baba	Store = NULL
State = 2	READ	Tape = aba	Store = NULL
State = 4	STORE	Tape = aba	Store = B
State = 2	READ	Tape = ba	Store = B
State = 3	STORE	Tape = ba	Store = BA
State = 2	READ	Tape = a	Store = BA
State = 4	STORE	Tape = a	Store = BAB
State = 2	READ	Tape = NULL	Store = BAB
State = 3	STORE	Tape = NULL	Store = BABA
State = 2	READ	Tape = NULL	Store = BABA
State = 5	CONSULT	Tape = NULL	Store = ABA
State = 7	SRC	Tape = NULL	Store = AAB
State = 9	CONSULT	Tape = NULL	Store = AB
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: babb

State = 1	START	Tape = babb	Store = NULL
State = 2	READ	Tape = abb	Store = NULL
State = 4	STORE	Tape = abb	Store = B
State = 2	READ	Tape = bb	Store = B
State = 3	STORE	Tape = bb	Store = BA
State = 2	READ	Tape = b	Store = BA
State = 4	STORE	Tape = b	Store = BAB
State = 2	READ	Tape = NULL	Store = BAB
State = 4	STORE	Tape = NULL	Store = BABB
State = 2	READ	Tape = NULL	Store = BABB
State = 5	CONSULT	Tape = NULL	Store = ABB
State = 7	SRC	Tape = NULL	Store = BAB
State = 9	CONSULT	Tape = NULL	Store = AB
State = 5	CONSULT	Tape = NULL	Store = B
State = 6	SRC	Tape = NULL	Store = B
State = 8	CONSULT	Tape = NULL	Store = NULL
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: bbaa

State = 1	START	Tape = bbaa	Store = NULL
State = 2	READ	Tape = baa	Store = NULL
State = 4	STORE	Tape = baa	Store = B
State = 2	READ	Tape = aa	Store = B
State = 4	STORE	Tape = aa	Store = BB
State = 2	READ	Tape = a	Store = BB
State = 3	STORE	Tape = a	Store = BBA
State = 2	READ	Tape = NULL	Store = BBA
State = 3	STORE	Tape = NULL	Store = BBAA
State = 2	READ	Tape = NULL	Store = BBAA
State = 5	CONSULT	Tape = NULL	Store = BAA
State = 7	SRC	Tape = NULL	Store = ABA
State = 9	CONSULT	Tape = NULL	Store = BA
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: bbab

State = 1	START	Tape = bbab	Store = NULL
State = 2	READ	Tape = bab	Store = NULL
State = 4	STORE	Tape = bab	Store = B
State = 2	READ	Tape = ab	Store = B
State = 4	STORE	Tape = ab	Store = BB
State = 2	READ	Tape = b	Store = BB
State = 3	STORE	Tape = b	Store = BBA
State = 2	READ	Tape = NULL	Store = BBA
State = 4	STORE	Tape = NULL	Store = BBAB
State = 2	READ	Tape = NULL	Store = BBAB

State =	5	CONSULT	Tape =	NULL	Store =	BAB
State =	7	SRC	Tape =	NULL	Store =	BBA
State =	9	CONSULT	Tape =	NULL	Store =	BA
State =	5	CONSULT	Tape =	NULL	Store =	A
State =	7	SRC	Tape =	NULL	Store =	A
State =	9	CONSULT	Tape =	NULL	Store =	NULL
State =	0	REJECT				

Enter word or 'N' for null string or '\*' to turn off machine: bbba

State =	1	START	Tape =	bbba	Store =	NULL
State =	2	READ	Tape =	bba	Store =	NULL
State =	4	STORE	Tape =	bba	Store =	B
State =	2	READ	Tape =	ba	Store =	B
State =	4	STORE	Tape =	ba	Store =	BB
State =	2	READ	Tape =	a	Store =	BB
State =	4	STORE	Tape =	a	Store =	BBB
State =	2	READ	Tape =	NULL	Store =	BBB
State =	3	STORE	Tape =	NULL	Store =	BBBA
State =	2	READ	Tape =	NULL	Store =	BBBA
State =	5	CONSULT	Tape =	NULL	Store =	BBA
State =	7	SRC	Tape =	NULL	Store =	ABB
State =	9	CONSULT	Tape =	NULL	Store =	BB
State =	0	REJECT				

Enter word or 'N' for null string or '\*' to turn off machine: bbbb

State =	1	START	Tape =	bbbb	Store =	NULL
State =	2	READ	Tape =	bbb	Store =	NULL
State =	4	STORE	Tape =	bbb	Store =	B
State =	2	READ	Tape =	bb	Store =	B
State =	4	STORE	Tape =	bb	Store =	BB
State =	2	READ	Tape =	b	Store =	BB
State =	4	STORE	Tape =	b	Store =	BBB
State =	2	READ	Tape =	NULL	Store =	BBB
State =	4	STORE	Tape =	NULL	Store =	BBBB
State =	2	READ	Tape =	NULL	Store =	BBBB
State =	5	CONSULT	Tape =	NULL	Store =	BBB
State =	7	SRC	Tape =	NULL	Store =	BBB
State =	9	CONSULT	Tape =	NULL	Store =	BB
State =	5	CONSULT	Tape =	NULL	Store =	B
State =	7	SRC	Tape =	NULL	Store =	B
State =	9	CONSULT	Tape =	NULL	Store =	NULL
State =	5	CONSULT	Tape =	NULL	Store =	NULL
State =	10	ACCEPT				

Enter word or 'N' for null string or '\*' to turn off machine: aaaaa

State =	1	START	Tape =	aaaaa	Store =	NULL
State =	2	READ	Tape =	aaaa	Store =	NULL
State =	3	STORE	Tape =	aaaa	Store =	A
State =	2	READ	Tape =	aaa	Store =	A
State =	3	STORE	Tape =	aaa	Store =	AA
State =	2	READ	Tape =	aa	Store =	AA
State =	3	STORE	Tape =	aa	Store =	AAA
State =	2	READ	Tape =	a	Store =	AAA
State =	3	STORE	Tape =	a	Store =	AAAA
State =	2	READ	Tape =	NULL	Store =	AAAA
State =	3	STORE	Tape =	NULL	Store =	AAAAA
State =	2	READ	Tape =	NULL	Store =	AAAAA
State =	5	CONSULT	Tape =	NULL	Store =	AAAA
State =	6	SRC	Tape =	NULL	Store =	AAAA
State =	8	CONSULT	Tape =	NULL	Store =	AAA
State =	5	CONSULT	Tape =	NULL	Store =	AA
State =	6	SRC	Tape =	NULL	Store =	AA

State = 8	CONSULT	Tape = NULL	Store = A
State = 5	CONSULT	Tape = NULL	Store = NULL
State = 6	SRC	Tape = NULL	Store = NULL
State = 8	CONSULT	Tape = NULL	Store = NULL
State = 10	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: aaaab

State = 1	START	Tape = aaaab	Store = NULL
State = 2	READ	Tape = aaab	Store = NULL
State = 3	STORE	Tape = aaab	Store = A
State = 2	READ	Tape = aab	Store = A
State = 3	STORE	Tape = aab	Store = AA
State = 2	READ	Tape = ab	Store = AA
State = 3	STORE	Tape = ab	Store = AAA
State = 2	READ	Tape = b	Store = AAA
State = 3	STORE	Tape = b	Store = AAAA
State = 2	READ	Tape = NULL	Store = AAAA
State = 4	STORE	Tape = NULL	Store = AAAAB
State = 2	READ	Tape = NULL	Store = AAAAB
State = 5	CONSULT	Tape = NULL	Store = AAAB
State = 6	SRC	Tape = NULL	Store = BAAA
State = 8	CONSULT	Tape = NULL	Store = AAA
State = 0	REJECT		

## -----Palindromes Continuation-----

Enter word or 'N' for null string or '\*' to turn off machine: aaaba

State = 1	START	Tape = aaaba	Store = NULL
State = 2	READ	Tape = aaba	Store = NULL
State = 3	STORE	Tape = aaba	Store = A
State = 2	READ	Tape = aba	Store = A
State = 3	STORE	Tape = aba	Store = AA
State = 2	READ	Tape = ba	Store = AA
State = 3	STORE	Tape = ba	Store = AAA
State = 2	READ	Tape = a	Store = AAA
State = 4	STORE	Tape = a	Store = AAAB
State = 2	READ	Tape = NULL	Store = AAAB
State = 3	STORE	Tape = NULL	Store = AAABA
State = 2	READ	Tape = NULL	Store = AAABA
State = 5	CONSULT	Tape = NULL	Store = AABA
State = 6	SRC	Tape = NULL	Store = AAAB
State = 8	CONSULT	Tape = NULL	Store = AAB
State = 5	CONSULT	Tape = NULL	Store = AB
State = 6	SRC	Tape = NULL	Store = BA
State = 8	CONSULT	Tape = NULL	Store = A
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: aaabb

State = 1	START	Tape = aaabb	Store = NULL
State = 2	READ	Tape = aabb	Store = NULL
State = 3	STORE	Tape = aabb	Store = A
State = 2	READ	Tape = abb	Store = A
State = 3	STORE	Tape = abb	Store = AA
State = 2	READ	Tape = bb	Store = AA
State = 3	STORE	Tape = bb	Store = AAA
State = 2	READ	Tape = b	Store = AAA
State = 4	STORE	Tape = b	Store = AAAB
State = 2	READ	Tape = NULL	Store = AAAB
State = 4	STORE	Tape = NULL	Store = AAABB
State = 2	READ	Tape = NULL	Store = AAABB
State = 5	CONSULT	Tape = NULL	Store = AABB
State = 6	SRC	Tape = NULL	Store = BAAB
State = 8	CONSULT	Tape = NULL	Store = AAB
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: aabaa

State = 1	START	Tape = aabaa	Store = NULL
State = 2	READ	Tape = abaa	Store = NULL
State = 3	STORE	Tape = abaa	Store = A
State = 2	READ	Tape = baa	Store = A
State = 3	STORE	Tape = baa	Store = AA
State = 2	READ	Tape = aa	Store = AA
State = 4	STORE	Tape = aa	Store = AAB
State = 2	READ	Tape = a	Store = AAB
State = 3	STORE	Tape = a	Store = AABA
State = 2	READ	Tape = NULL	Store = AABA
State = 3	STORE	Tape = NULL	Store = AABAA
State = 2	READ	Tape = NULL	Store = AABAA
State = 5	CONSULT	Tape = NULL	Store = ABAA
State = 6	SRC	Tape = NULL	Store = AABA
State = 8	CONSULT	Tape = NULL	Store = ABA
State = 5	CONSULT	Tape = NULL	Store = BA
State = 6	SRC	Tape = NULL	Store = AB
State = 8	CONSULT	Tape = NULL	Store = B

```

State = 5 CONSULT    Tape = NULL    Store = NULL
State = 7 SRC        Tape = NULL    Store = NULL
State = 9 CONSULT    Tape = NULL    Store = NULL
State = 10 ACCEPT

```

Enter word or 'N' for null string or '\*' to turn off machine: aabab

```

State = 1 START      Tape = aabab    Store = NULL
State = 2 READ       Tape = abab     Store = NULL
State = 3 STORE      Tape = abab     Store = A
State = 2 READ       Tape = bab      Store = A
State = 3 STORE      Tape = bab      Store = AA
State = 2 READ       Tape = ab       Store = AA
State = 4 STORE      Tape = ab       Store = AAB
State = 2 READ       Tape = b        Store = AAB
State = 3 STORE      Tape = b        Store = AABA
State = 2 READ       Tape = NULL     Store = AABA
State = 4 STORE      Tape = NULL     Store = AABAB
State = 2 READ       Tape = NULL     Store = AABAB
State = 5 CONSULT    Tape = NULL     Store = ABAB
State = 6 SRC        Tape = NULL     Store = BABA
State = 8 CONSULT    Tape = NULL     Store = ABA
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aabba

```

State = 1 START      Tape = aabba    Store = NULL
State = 2 READ       Tape = abba     Store = NULL
State = 3 STORE      Tape = abba     Store = A
State = 2 READ       Tape = bba      Store = A
State = 3 STORE      Tape = bba      Store = AA
State = 2 READ       Tape = ba       Store = AA
State = 4 STORE      Tape = ba       Store = AAB
State = 2 READ       Tape = a        Store = AAB
State = 4 STORE      Tape = a        Store = AABB
State = 2 READ       Tape = NULL     Store = AABB
State = 3 STORE      Tape = NULL     Store = AABBA
State = 2 READ       Tape = NULL     Store = AABBA
State = 5 CONSULT    Tape = NULL     Store = ABBA
State = 6 SRC        Tape = NULL     Store = AABBB
State = 8 CONSULT    Tape = NULL     Store = ABB
State = 5 CONSULT    Tape = NULL     Store = BB
State = 6 SRC        Tape = NULL     Store = BB
State = 8 CONSULT    Tape = NULL     Store = B
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: aabbb

```

State = 1 START      Tape = aabbb    Store = NULL
State = 2 READ       Tape = abbb     Store = NULL
State = 3 STORE      Tape = abbb     Store = A
State = 2 READ       Tape = bbb      Store = A
State = 3 STORE      Tape = bbb      Store = AA
State = 2 READ       Tape = bb       Store = AA
State = 4 STORE      Tape = bb       Store = AAB
State = 2 READ       Tape = b        Store = AAB
State = 4 STORE      Tape = b        Store = AABB
State = 2 READ       Tape = NULL     Store = AABB
State = 4 STORE      Tape = NULL     Store = AABBB
State = 2 READ       Tape = NULL     Store = AABBB
State = 5 CONSULT    Tape = NULL     Store = ABBBB
State = 6 SRC        Tape = NULL     Store = BABBB
State = 8 CONSULT    Tape = NULL     Store = ABB
State = 0 REJECT

```

Enter word or 'N' for null string or '\*' to turn off machine: abaaa

State = 1	START	Tape = abaaa	Store = NULL
State = 2	READ	Tape = baaa	Store = NULL
State = 3	STORE	Tape = baaa	Store = A
State = 2	READ	Tape = aaa	Store = A
State = 4	STORE	Tape = aaa	Store = AB
State = 2	READ	Tape = aa	Store = AB
State = 3	STORE	Tape = aa	Store = ABA
State = 2	READ	Tape = a	Store = ABA
State = 3	STORE	Tape = a	Store = ABAA
State = 2	READ	Tape = NULL	Store = ABAA
State = 3	STORE	Tape = NULL	Store = ABAAA
State = 2	READ	Tape = NULL	Store = ABAAA
State = 5	CONSULT	Tape = NULL	Store = BAAA
State = 6	SRC	Tape = NULL	Store = ABAA
State = 8	CONSULT	Tape = NULL	Store = BAA
State = 5	CONSULT	Tape = NULL	Store = AA
State = 7	SRC	Tape = NULL	Store = AA
State = 9	CONSULT	Tape = NULL	Store = A
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: abaab

State = 1	START	Tape = abaab	Store = NULL
State = 2	READ	Tape = baab	Store = NULL
State = 3	STORE	Tape = baab	Store = A
State = 2	READ	Tape = aab	Store = A
State = 4	STORE	Tape = aab	Store = AB
State = 2	READ	Tape = ab	Store = AB
State = 3	STORE	Tape = ab	Store = ABA
State = 2	READ	Tape = b	Store = ABA
State = 3	STORE	Tape = b	Store = ABAA
State = 2	READ	Tape = NULL	Store = ABAA
State = 4	STORE	Tape = NULL	Store = ABAAAB
State = 2	READ	Tape = NULL	Store = ABAAAB
State = 5	CONSULT	Tape = NULL	Store = BAAB
State = 6	SRC	Tape = NULL	Store = BBAA
State = 8	CONSULT	Tape = NULL	Store = BAA
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: ababa

State = 1	START	Tape = ababa	Store = NULL
State = 2	READ	Tape = baba	Store = NULL
State = 3	STORE	Tape = baba	Store = A
State = 2	READ	Tape = aba	Store = A
State = 4	STORE	Tape = aba	Store = AB
State = 2	READ	Tape = ba	Store = AB
State = 3	STORE	Tape = ba	Store = ABA
State = 2	READ	Tape = a	Store = ABA
State = 4	STORE	Tape = a	Store = ABAB
State = 2	READ	Tape = NULL	Store = ABAB
State = 3	STORE	Tape = NULL	Store = ABABA
State = 2	READ	Tape = NULL	Store = ABABA
State = 5	CONSULT	Tape = NULL	Store = BABA
State = 6	SRC	Tape = NULL	Store = ABAB
State = 8	CONSULT	Tape = NULL	Store = BAB
State = 5	CONSULT	Tape = NULL	Store = AB
State = 7	SRC	Tape = NULL	Store = BA
State = 9	CONSULT	Tape = NULL	Store = A
State = 5	CONSULT	Tape = NULL	Store = NULL
State = 6	SRC	Tape = NULL	Store = NULL
State = 8	CONSULT	Tape = NULL	Store = NULL
State = 10	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: ababb

State =	1	START	Tape =	ababb	Store =	NULL
State =	2	READ	Tape =	babb	Store =	NULL
State =	3	STORE	Tape =	babb	Store =	A
State =	2	READ	Tape =	abb	Store =	A
State =	4	STORE	Tape =	abb	Store =	AB
State =	2	READ	Tape =	bb	Store =	AB
State =	3	STORE	Tape =	bb	Store =	ABA
State =	2	READ	Tape =	b	Store =	ABA
State =	4	STORE	Tape =	b	Store =	ABAB
State =	2	READ	Tape =	NULL	Store =	ABAB
State =	4	STORE	Tape =	NULL	Store =	ABABB
State =	2	READ	Tape =	NULL	Store =	ABABB
State =	5	CONSULT	Tape =	NULL	Store =	BABB
State =	6	SRC	Tape =	NULL	Store =	BBAB
State =	8	CONSULT	Tape =	NULL	Store =	BAB
State =	0	REJECT				

Enter word or 'N' for null string or '\*' to turn off machine: abbba

State =	1	START	Tape =	abbba	Store =	NULL
State =	2	READ	Tape =	bbba	Store =	NULL
State =	3	STORE	Tape =	bbba	Store =	A
State =	2	READ	Tape =	baa	Store =	A
State =	4	STORE	Tape =	baa	Store =	AB
State =	2	READ	Tape =	aa	Store =	AB
State =	4	STORE	Tape =	aa	Store =	ABB
State =	2	READ	Tape =	a	Store =	ABB
State =	3	STORE	Tape =	a	Store =	ABBA
State =	2	READ	Tape =	NULL	Store =	ABBA
State =	3	STORE	Tape =	NULL	Store =	ABBAA
State =	2	READ	Tape =	NULL	Store =	ABBAA
State =	5	CONSULT	Tape =	NULL	Store =	BBAA
State =	6	SRC	Tape =	NULL	Store =	ABBA
State =	8	CONSULT	Tape =	NULL	Store =	BBA
State =	5	CONSULT	Tape =	NULL	Store =	BA
State =	7	SRC	Tape =	NULL	Store =	AB
State =	9	CONSULT	Tape =	NULL	Store =	B
State =	0	REJECT				

Enter word or 'N' for null string or '\*' to turn off machine: abbab

State =	1	START	Tape =	abbab	Store =	NULL
State =	2	READ	Tape =	bbab	Store =	NULL
State =	3	STORE	Tape =	bbab	Store =	A
State =	2	READ	Tape =	bab	Store =	A
State =	4	STORE	Tape =	bab	Store =	AB
State =	2	READ	Tape =	ab	Store =	AB
State =	4	STORE	Tape =	ab	Store =	ABB
State =	2	READ	Tape =	b	Store =	ABB
State =	3	STORE	Tape =	b	Store =	ABBA
State =	2	READ	Tape =	NULL	Store =	ABBA
State =	4	STORE	Tape =	NULL	Store =	ABBAB
State =	2	READ	Tape =	NULL	Store =	ABBAB
State =	5	CONSULT	Tape =	NULL	Store =	BBAB
State =	6	SRC	Tape =	NULL	Store =	BBBA
State =	8	CONSULT	Tape =	NULL	Store =	BBA
State =	0	REJECT				

Enter word or 'N' for null string or '\*' to turn off machine: abbba

State =	1	START	Tape =	abbba	Store =	NULL
State =	2	READ	Tape =	bbba	Store =	NULL



State = 3	STORE	Tape = bbba	Store = A
State = 2	READ	Tape = bba	Store = A
State = 4	STORE	Tape = bba	Store = AB
State = 2	READ	Tape = ba	Store = AB
State = 4	STORE	Tape = ba	Store = ABB
State = 2	READ	Tape = a	Store = ABB
State = 4	STORE	Tape = a	Store = ABBB
State = 2	READ	Tape = NULL	Store = ABBB
State = 3	STORE	Tape = NULL	Store = ABBSA
State = 2	READ	Tape = NULL	Store = ABBSA
State = 5	CONSULT	Tape = NULL	Store = BSBA
State = 6	SRC	Tape = NULL	Store = ABBS
State = 8	CONSULT	Tape = NULL	Store = BS
State = 5	CONSULT	Tape = NULL	Store = BS
State = 7	SRC	Tape = NULL	Store = BS
State = 9	CONSULT	Tape = NULL	Store = B
State = 5	CONSULT	Tape = NULL	Store = NULL
State = 7	SRC	Tape = NULL	Store = NULL
State = 9	CONSULT	Tape = NULL	Store = NULL
State = 10	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: abbbb

State = 1	START	Tape = abbbb	Store = NULL
State = 2	READ	Tape = bbbb	Store = NULL
State = 3	STORE	Tape = bbbb	Store = A
State = 2	READ	Tape = bbb	Store = A
State = 4	STORE	Tape = bbb	Store = AB
State = 2	READ	Tape = bb	Store = AB
State = 4	STORE	Tape = bb	Store = ABB
State = 2	READ	Tape = b	Store = ABB
State = 4	STORE	Tape = b	Store = ABBS
State = 2	READ	Tape = NULL	Store = ABBS
State = 4	STORE	Tape = NULL	Store = ABBSA
State = 2	READ	Tape = NULL	Store = ABBSA
State = 5	CONSULT	Tape = NULL	Store = BSBS
State = 6	SRC	Tape = NULL	Store = BSBS
State = 8	CONSULT	Tape = NULL	Store = BS
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: baaaa

State = 1	START	Tape = baaaa	Store = NULL
State = 2	READ	Tape = aaaa	Store = NULL
State = 4	STORE	Tape = aaaa	Store = B
State = 2	READ	Tape = aaa	Store = B
State = 3	STORE	Tape = aaa	Store = BA
State = 2	READ	Tape = aa	Store = BA
State = 3	STORE	Tape = aa	Store = BAA
State = 2	READ	Tape = a	Store = BAA
State = 3	STORE	Tape = a	Store = BAAA
State = 2	READ	Tape = NULL	Store = BAAA
State = 3	STORE	Tape = NULL	Store = BAAAA
State = 2	READ	Tape = NULL	Store = BAAAA
State = 5	CONSULT	Tape = NULL	Store = AAAA
State = 7	SRC	Tape = NULL	Store = AAAA
State = 9	CONSULT	Tape = NULL	Store = AAA
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: baaab

State = 1	START	Tape = baaab	Store = NULL
State = 2	READ	Tape = aaab	Store = NULL
State = 4	STORE	Tape = aaab	Store = B
State = 2	READ	Tape = aab	Store = B

State = 3	STORE	Tape = aab	Store = BA
State = 2	READ	Tape = ab	Store = BA
State = 3	STORE	Tape = ab	Store = BAA
State = 2	READ	Tape = b	Store = BAA
State = 3	STORE	Tape = b	Store = BAAA
State = 2	READ	Tape = NULL	Store = BAAA
State = 4	STORE	Tape = NULL	Store = BAAAB
State = 2	READ	Tape = NULL	Store = BAAAB
State = 5	CONSULT	Tape = NULL	Store = AAAB
State = 7	SRC	Tape = NULL	Store = BAAA
State = 9	CONSULT	Tape = NULL	Store = AAA
State = 5	CONSULT	Tape = NULL	Store = AA
State = 6	SRC	Tape = NULL	Store = AA
State = 8	CONSULT	Tape = NULL	Store = A
State = 5	CONSULT	Tape = NULL	Store = NULL
State = 6	SRC	Tape = NULL	Store = NULL
State = 8	CONSULT	Tape = NULL	Store = NULL
State = 10	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: baaba

State = 1	START	Tape = baaba	Store = NULL
State = 2	READ	Tape = aaba	Store = NULL
State = 4	STORE	Tape = aaba	Store = B
State = 2	READ	Tape = aba	Store = B
State = 3	STORE	Tape = aba	Store = BA
State = 2	READ	Tape = ba	Store = BA
State = 3	STORE	Tape = ba	Store = BAA
State = 2	READ	Tape = a	Store = BAA
State = 4	STORE	Tape = a	Store = BAAB
State = 2	READ	Tape = NULL	Store = BAAB
State = 3	STORE	Tape = NULL	Store = BAABA
State = 2	READ	Tape = NULL	Store = BAABA
State = 5	CONSULT	Tape = NULL	Store = AABA
State = 7	SRC	Tape = NULL	Store = AAAB
State = 9	CONSULT	Tape = NULL	Store = AAB
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: baabb

State = 1	START	Tape = baabb	Store = NULL
State = 2	READ	Tape = aabb	Store = NULL
State = 4	STORE	Tape = aabb	Store = B
State = 2	READ	Tape = abb	Store = B
State = 3	STORE	Tape = abb	Store = BA
State = 2	READ	Tape = bb	Store = BA
State = 3	STORE	Tape = bb	Store = BAA
State = 2	READ	Tape = b	Store = BAA
State = 4	STORE	Tape = b	Store = BAAB
State = 2	READ	Tape = NULL	Store = BAAB
State = 4	STORE	Tape = NULL	Store = BAABB
State = 2	READ	Tape = NULL	Store = BAABB
State = 5	CONSULT	Tape = NULL	Store = AABB
State = 7	SRC	Tape = NULL	Store = BAAB
State = 9	CONSULT	Tape = NULL	Store = AAB
State = 5	CONSULT	Tape = NULL	Store = AB
State = 6	SRC	Tape = NULL	Store = BA
State = 8	CONSULT	Tape = NULL	Store = A
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: babaa

State = 1	START	Tape = babaa	Store = NULL
State = 2	READ	Tape = abaa	Store = NULL
State = 4	STORE	Tape = abaa	Store = B

State = 2	READ	Tape = baa	Store = B
State = 3	STORE	Tape = baa	Store = BA
State = 2	READ	Tape = aa	Store = BA
State = 4	STORE	Tape = aa	Store = BAB
State = 2	READ	Tape = a	Store = BAB
State = 3	STORE	Tape = a	Store = BABA
State = 2	READ	Tape = NULL	Store = BABA
State = 3	STORE	Tape = NULL	Store = BABAA
State = 2	READ	Tape = NULL	Store = BABAA
State = 5	CONSULT	Tape = NULL	Store = ABAA
State = 7	SRC	Tape = NULL	Store = AABA
State = 9	CONSULT	Tape = NULL	Store = ABA
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: babab

State = 1	START	Tape = babab	Store = NULL
State = 2	READ	Tape = abab	Store = NULL
State = 4	STORE	Tape = abab	Store = B
State = 2	READ	Tape = bab	Store = B
State = 3	STORE	Tape = bab	Store = BA
State = 2	READ	Tape = ab	Store = BA
State = 4	STORE	Tape = ab	Store = BAB
State = 2	READ	Tape = b	Store = BAB
State = 3	STORE	Tape = b	Store = BABA
State = 2	READ	Tape = NULL	Store = BABA
State = 4	STORE	Tape = NULL	Store = BABAB
State = 2	READ	Tape = NULL	Store = BABAB
State = 5	CONSULT	Tape = NULL	Store = ABAB
State = 7	SRC	Tape = NULL	Store = BABA
State = 9	CONSULT	Tape = NULL	Store = ABA
State = 5	CONSULT	Tape = NULL	Store = BA
State = 6	SRC	Tape = NULL	Store = AB
State = 8	CONSULT	Tape = NULL	Store = B
State = 5	CONSULT	Tape = NULL	Store = NULL
State = 7	SRC	Tape = NULL	Store = NULL
State = 9	CONSULT	Tape = NULL	Store = NULL
State = 10	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: babba

State = 1	START	Tape = babba	Store = NULL
State = 2	READ	Tape = abba	Store = NULL
State = 4	STORE	Tape = abba	Store = B
State = 2	READ	Tape = bba	Store = B
State = 3	STORE	Tape = bba	Store = BA
State = 2	READ	Tape = ba	Store = BA
State = 4	STORE	Tape = ba	Store = BAB
State = 2	READ	Tape = a	Store = BAB
State = 4	STORE	Tape = a	Store = BABB
State = 2	READ	Tape = NULL	Store = BABB
State = 3	STORE	Tape = NULL	Store = BABBA
State = 2	READ	Tape = NULL	Store = BABBA
State = 5	CONSULT	Tape = NULL	Store = ABBA
State = 7	SRC	Tape = NULL	Store = AABB
State = 9	CONSULT	Tape = NULL	Store = ABB
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: babbb

State = 1	START	Tape = babbb	Store = NULL
State = 2	READ	Tape = abbb	Store = NULL
State = 4	STORE	Tape = abbb	Store = B
State = 2	READ	Tape = bbb	Store = B
State = 3	STORE	Tape = bbb	Store = BA

State = 2	READ	Tape = bb	Store = BA
State = 4	STORE	Tape = bb	Store = BAB
State = 2	READ	Tape = b	Store = BAB
State = 4	STORE	Tape = b	Store = BABB
State = 2	READ	Tape = NULL	Store = BABB
State = 4	STORE	Tape = NULL	Store = BABBB
State = 2	READ	Tape = NULL	Store = BABBB
State = 5	CONSULT	Tape = NULL	Store = ABBB
State = 7	SRC	Tape = NULL	Store = BABB
State = 9	CONSULT	Tape = NULL	Store = ABB
State = 5	CONSULT	Tape = NULL	Store = BB
State = 6	SRC	Tape = NULL	Store = BB
State = 8	CONSULT	Tape = NULL	Store = B
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: bbaaa

State = 1	START	Tape = bbaaa	Store = NULL
State = 2	READ	Tape = baaa	Store = NULL
State = 4	STORE	Tape = baaa	Store = B
State = 2	READ	Tape = aaa	Store = B
State = 4	STORE	Tape = aaa	Store = BB
State = 2	READ	Tape = aa	Store = BB
State = 3	STORE	Tape = aa	Store = BBA
State = 2	READ	Tape = a	Store = BBA
State = 3	STORE	Tape = a	Store = BBAA
State = 2	READ	Tape = NULL	Store = BBAA
State = 3	STORE	Tape = NULL	Store = BBAAA
State = 2	READ	Tape = NULL	Store = BBAAA
State = 5	CONSULT	Tape = NULL	Store = BAAA
State = 7	SRC	Tape = NULL	Store = ABAA
State = 9	CONSULT	Tape = NULL	Store = BAA
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: bbaab

State = 1	START	Tape = bbaab	Store = NULL
State = 2	READ	Tape = baab	Store = NULL
State = 4	STORE	Tape = baab	Store = B
State = 2	READ	Tape = aab	Store = B
State = 4	STORE	Tape = aab	Store = BB
State = 2	READ	Tape = ab	Store = BB
State = 3	STORE	Tape = ab	Store = BBA
State = 2	READ	Tape = b	Store = BBA
State = 3	STORE	Tape = b	Store = BBAA
State = 2	READ	Tape = NULL	Store = BBAA
State = 4	STORE	Tape = NULL	Store = BBAAAB
State = 2	READ	Tape = NULL	Store = BBAAAB
State = 5	CONSULT	Tape = NULL	Store = BAAB
State = 7	SRC	Tape = NULL	Store = BBAA
State = 9	CONSULT	Tape = NULL	Store = BAA
State = 5	CONSULT	Tape = NULL	Store = AA
State = 7	SRC	Tape = NULL	Store = AA
State = 9	CONSULT	Tape = NULL	Store = A
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: bbaba

State = 1	START	Tape = bbaba	Store = NULL
State = 2	READ	Tape = baba	Store = NULL
State = 4	STORE	Tape = baba	Store = B
State = 2	READ	Tape = aba	Store = B
State = 4	STORE	Tape = aba	Store = BB
State = 2	READ	Tape = ba	Store = BB
State = 3	STORE	Tape = ba	Store = BBA

State = 2	READ	Tape = a	Store = BBA
State = 4	STORE	Tape = a	Store = BBAB
State = 2	READ	Tape = NULL	Store = BBAB
State = 3	STORE	Tape = NULL	Store = BBABA
State = 2	READ	Tape = NULL	Store = BBABA
State = 5	CONSULT	Tape = NULL	Store = BABA
State = 7	SRC	Tape = NULL	Store = ABAB
State = 9	CONSULT	Tape = NULL	Store = BAB
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: bbabb

State = 1	START	Tape = bbabb	Store = NULL
State = 2	READ	Tape = babb	Store = NULL
State = 4	STORE	Tape = babb	Store = B
State = 2	READ	Tape = abb	Store = B
State = 4	STORE	Tape = abb	Store = BB
State = 2	READ	Tape = bb	Store = BB
State = 3	STORE	Tape = bb	Store = BBA
State = 2	READ	Tape = b	Store = BBA
State = 4	STORE	Tape = b	Store = BBAB
State = 2	READ	Tape = NULL	Store = BBAB
State = 4	STORE	Tape = NULL	Store = BBABB
State = 2	READ	Tape = NULL	Store = BBABB
State = 5	CONSULT	Tape = NULL	Store = BABB
State = 7	SRC	Tape = NULL	Store = BBAB
State = 9	CONSULT	Tape = NULL	Store = BAB
State = 5	CONSULT	Tape = NULL	Store = AB
State = 7	SRC	Tape = NULL	Store = BA
State = 9	CONSULT	Tape = NULL	Store = A
State = 5	CONSULT	Tape = NULL	Store = NULL
State = 6	SRC	Tape = NULL	Store = NULL
State = 8	CONSULT	Tape = NULL	Store = NULL
State = 10	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: bbbaa

State = 1	START	Tape = bbbaa	Store = NULL
State = 2	READ	Tape = bbbaa	Store = NULL
State = 4	STORE	Tape = bbbaa	Store = B
State = 2	READ	Tape = baa	Store = B
State = 4	STORE	Tape = baa	Store = BB
State = 2	READ	Tape = aa	Store = BB
State = 4	STORE	Tape = aa	Store = BBB
State = 2	READ	Tape = a	Store = BBB
State = 3	STORE	Tape = a	Store = BBBA
State = 2	READ	Tape = NULL	Store = BBBA
State = 3	STORE	Tape = NULL	Store = BBBAA
State = 2	READ	Tape = NULL	Store = BBBAA
State = 5	CONSULT	Tape = NULL	Store = BBAA
State = 7	SRC	Tape = NULL	Store = ABBA
State = 9	CONSULT	Tape = NULL	Store = BBA
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: bbbab

State = 1	START	Tape = bbbab	Store = NULL
State = 2	READ	Tape = bbbab	Store = NULL
State = 4	STORE	Tape = bbbab	Store = B
State = 2	READ	Tape = bab	Store = B
State = 4	STORE	Tape = bab	Store = BB
State = 2	READ	Tape = ab	Store = BB
State = 4	STORE	Tape = ab	Store = BBB
State = 2	READ	Tape = b	Store = BBB
State = 3	STORE	Tape = b	Store = BBBA

State = 2	READ	Tape = NULL	Store = BBBA
State = 4	STORE	Tape = NULL	Store = BBBAB
State = 2	READ	Tape = NULL	Store = BBBAB
State = 5	CONSULT	Tape = NULL	Store = BBAB
State = 7	SRC	Tape = NULL	Store = BBBA
State = 9	CONSULT	Tape = NULL	Store = BBA
State = 5	CONSULT	Tape = NULL	Store = BA
State = 7	SRC	Tape = NULL	Store = AB
State = 9	CONSULT	Tape = NULL	Store = B
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: bbbba

State = 1	START	Tape = bbbba	Store = NULL
State = 2	READ	Tape = bbba	Store = NULL
State = 4	STORE	Tape = bbba	Store = B
State = 2	READ	Tape = bba	Store = B
State = 4	STORE	Tape = bba	Store = BB
State = 2	READ	Tape = ba	Store = BB
State = 4	STORE	Tape = ba	Store = BBB
State = 2	READ	Tape = a	Store = BBB
State = 4	STORE	Tape = a	Store = BBBB
State = 2	READ	Tape = NULL	Store = BBBB
State = 3	STORE	Tape = NULL	Store = BBBBA
State = 2	READ	Tape = NULL	Store = BBBBA
State = 5	CONSULT	Tape = NULL	Store = BBBA
State = 7	SRC	Tape = NULL	Store = ABBB
State = 9	CONSULT	Tape = NULL	Store = BBB
State = 0	REJECT		

Enter word or 'N' for null string or '\*' to turn off machine: bbbba

State = 1	START	Tape = bbbba	Store = NULL
State = 2	READ	Tape = bbba	Store = NULL
State = 4	STORE	Tape = bbba	Store = B
State = 2	READ	Tape = bba	Store = B
State = 4	STORE	Tape = bba	Store = BB
State = 2	READ	Tape = ba	Store = BB
State = 4	STORE	Tape = ba	Store = BBB
State = 2	READ	Tape = a	Store = BBB
State = 4	STORE	Tape = a	Store = BBBB
State = 2	READ	Tape = NULL	Store = BBBB
State = 3	STORE	Tape = NULL	Store = BBBBA
State = 2	READ	Tape = NULL	Store = BBBBA
State = 5	CONSULT	Tape = NULL	Store = BBBA
State = 7	SRC	Tape = NULL	Store = ABBB
State = 9	CONSULT	Tape = NULL	Store = BBB
State = 0	REJECT		
State = 9	CONSULT	Tape = NULL	Store = NULL
State = 10	ACCEPT		

Enter word or 'N' for null string or '\*' to turn off machine: bbbbb

State = 1	START	Tape = bbbbb	Store = NULL
State = 2	READ	Tape = bbbb	Store = NULL
State = 4	STORE	Tape = bbbb	Store = B
State = 2	READ	Tape = bbb	Store = B
State = 4	STORE	Tape = bbb	Store = BB
State = 2	READ	Tape = bb	Store = BB
State = 4	STORE	Tape = bb	Store = BBB
State = 2	READ	Tape = b	Store = BBB
State = 4	STORE	Tape = b	Store = BBBB
State = 2	READ	Tape = NULL	Store = BBBB
State = 4	STORE	Tape = NULL	Store = BBBBB
State = 2	READ	Tape = NULL	Store = BBBBB

State = 5 CONSULT	Tape = NULL	Store = BBBB
State = 7 SRC	Tape = NULL	Store = BBBB
State = 9 CONSULT	Tape = NULL	Store = BBB
State = 5 CONSULT	Tape = NULL	Store = BB
State = 7 SRC	Tape = NULL	Store = BB
State = 9 CONSULT	Tape = NULL	Store = B
State = 5 CONSULT	Tape = NULL	Store = NULL
State = 7 SRC	Tape = NULL	Store = NULL
State = 9 CONSULT	Tape = NULL	Store = NULL
State = 10 ACCEPT		

If you are still not satisfy...YOU ARE A REAL MASOCHIST! But, if you happen to be a lady, can I have your phone number? ;-)

## Post Machine: INDEX



Ronald Eggers [ronald.eggert@gte.net](mailto:ronald.eggert@gte.net)  
Qian Chang [qchang@dhvx20.csudh.edu](mailto:qchang@dhvx20.csudh.edu)  
Mando Gomez [mandolux@yahoo.com](mailto:mandolux@yahoo.com)

Question: Do you think this article needs an index to navigate? I don't think so! The table of contents should be enough....

#	N
A	O
B	P
C	Q
D	R
E	S
F	T
G	U
H	V
I	W
J	X
K	Y
L	Z
M	



## Post Machine: Contact Information:



Ronald Eggers <a href="mailto:ronald.eggert@gte.net">ronald.eggert@gte.net</a> Qian Chang <a href="mailto:qchang@dhvx20.csudh.edu">qchang@dhvx20.csudh.edu</a> Mando Gomez <a href="mailto:mandolux@yahoo.com">mandolux@yahoo.com</a>
---

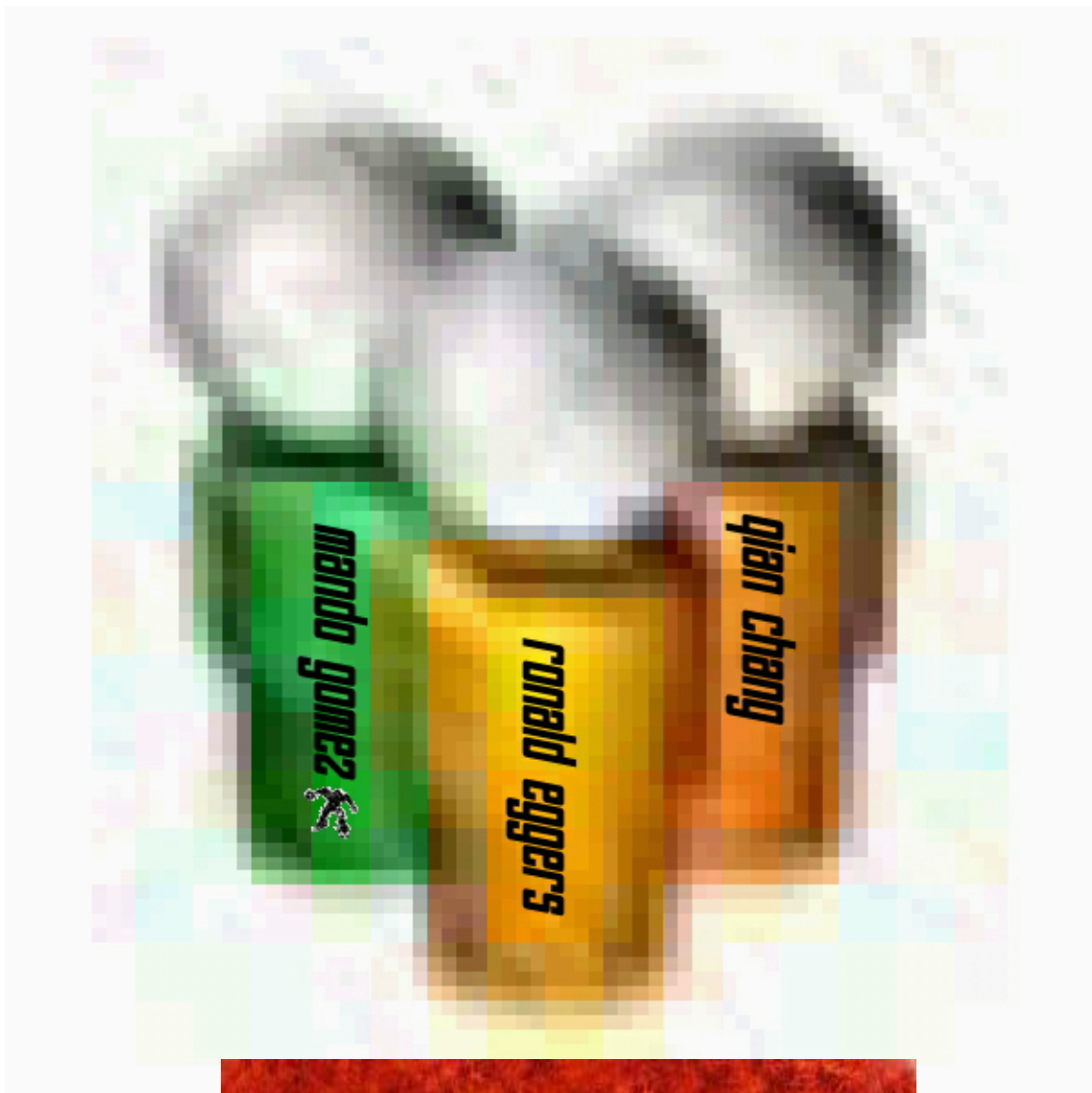
---

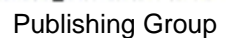
...we like to be contacted by email...

Ronald Eggers <the big kahuna> [ronald.eggert@gte.net](mailto:ronald.eggert@gte.net)  
Qian Chang <the little kahuna> [qchang@dhvx20.csudh.edu](mailto:qchang@dhvx20.csudh.edu)  
Mando Gomez <theDude> [mandolux@yahoo.com](mailto:mandolux@yahoo.com)

Or visit our web pages at:

<http://www.geocities.com/mandolux>  
<http://csc208.csudh.edu/agomez>  
etc...



[illegible]

YES! This *is* the last  
page....