



E.T.S.I. Técnicos en Informática de Sistemas
Laboratorio de Programación II. 1ºB
Práctica 6: Estructuras de Datos Avanzadas (2)
2000/2001

Codificando en lenguaje ensamblador

Supongamos que tenemos una máquina con un solo registro y seis instrucciones:

LD	A	Sitúa el operando A en el registro
ST	A	Sitúa el contenido del registro en A
AD	A	Suma A al contenido del registro
SB	A	Resta A del contenido del registro
ML	A	Multiplica el contenido del registro por A
DV	A	Divide el contenido del registro por A

Escribe un programa que acepte una expresión en postfija que contenga operandos simples (formados por letras mayúsculas) y operadores (+, -, *, /), y que escriba en la pantalla la secuencia de instrucciones del tipo anterior para evaluar la expresión en dicha máquina, dejando el resultado en el registro. Por ejemplo, dada la expresión: $ABC*+DE-/$, el programa debería dar como resultado:

```
LD B
ML C
ST Temp1
LD A
AD Temp1
ST Temp2
LD D
SB E
ST Temp3
LD Temp2
DV Temp3
ST Temp4
LD Temp4
```

Para realizar este ejercicio debes utilizar una Pila. Por ello debes implementar un TAD Pila. El programa debe ser independiente de que la Pila haya sido implementada estática o dinámicamente por lo que deberá tener en cuenta si la Pila se llena o no.

Consideraciones:

- (1) Tratamiento de errores: se "engordara" cada procedimiento (del TAD Pila) con variables de tipo Booleano (o de tipo enumerado si se quiere) que serán TRUE si el procedimiento se ejecuto sin problemas y FALSE en otro caso. Ver prácticas anteriores.
- (2) Si hay operaciones sobre las cadenas están se deben modularizar en un módulo para las cadenas.
- (3) Recuerde que en los procedimientos auxiliares de un TAD habrán de aparecer únicamente en el módulo de implementación. No defina más de un tipo abstracto por módulo.
- (4) No se puede usar las funciones de la librería STR y recuerde que para aprobar hay que modularizar (uso de procedimientos y funciones).
- (5) Se debe evitar el uso de variables globales en los procedimientos del programa principal.