

Ejercicio 1. c

ALGORITMO BorraMaximo(VAR lista: TipoPuntero)

VARIABLES

ptr,anterior,maximo:TipoPuntero

INICIO

SI (lista <>NIL) ENTONCES (* si lista = NIL no hay nada que borrar *)

maximo:=lista

ptr:=lista

MIENTRAS (ptr^.sig<>NIL) HACER

SI ptr^.sig^.dato>maximo^.dato ENTONCES

anterior:=ptr

maximo:=ptr^.sig

FINSI

ptr:=ptr^.sig

FINMIENTRAS

SI maximo=lista ENTONCES (* El máximo es el 1º de los nodos *)

lista:=lista^.sig

ENOTROCASO (* El máximo es un nodo intermedio *)

anterior^.sig:=maximo^.sig

FINSI

LIBERAR(maximo,TAMAÑO(Nodo))

FINSI

FIN

Ejercicio 1.d

ALGORITMO IntercambiarValorNésimoConMésimo

(VAR lista: TipoPuntero;n,m:N)

VARIABLES

PtrN,PtrM:TipoPuntero

temp:N

INICIO

SI (m<>n) AND (n>0) AND (m>0) ENTONCES

SI n>m ENTONCES (* Para mejorar la eficiencia *)

temp:=n

n:=m

m:=temp

FINSI (* siempre m > n *)

PtrN:=lista

MIENTRAS (PtrN<>NIL) AND (n>1) HACER

PtrN:=PtrN^.sig

n:=n-1

m:=m-1

FINMIENTRAS

PtrM:=PtrN

MIENTRAS (PtrM<>NIL) AND (m>1) HACER

PtrM:=PtrM^.sig

m:=m-1

FINMIENTRAS

SI (PtrN<>NIL) AND (PtrM<>NIL) ENTONCES

temp:=PtrN^.dato

PtrN^.dato:=PtrM^.dato

PtrM^.dato:=temp

FINSI

FINSI

FIN

Ejercicio 1.e

ALGORITMO Concatenar(VAR lista1, lista2: TipoPuntero)

(* Solución Iterativa *)

VARIABLES

ptr: TipoPuntero

INICIO

SI lista2 <> NIL ENTONCES

SI lista1 = NIL ENTONCES

lista1 := lista2

lista2 := NIL

ENOTROCASO

ptr := lista1

MIENTRAS (ptr^.sig <> NIL) HACER

ptr := ptr^.sig

FINMIENTRAS

ptr^.sig := lista2

lista2 := NIL

FINSI

FINSI

FIN

Ejercicio 1.e

ALGORITMO Concatenar(VAR lista1,lista2: TipoPuntero)

(* Solución Recursiva *)

INICIO

SI lista2<>NIL ENTONCES

SI lista1=NIL ENTONCES

lista1:=lista2

lista2:=NIL

ENOTROCASO

Concatenar(lista1^.sig,lista2)

FINSI

FINSI

FIN

Ejercicio 2 (b)

ALGORITMO Purga(VAR L:TipoLista)

(* Solución Iterativa *)

VARIABLES

Ptr:TipoLista

INICIO

Ptr:=L

MIENTRAS (Ptr<>NIL) HACER

BorraTodasOcurrencias(Ptr^.valor,Ptr^.sig)

Ptr:=Ptr^.sig

FINMIENTRAS

FIN

Ejercicio 2 (b)

ALGORITMO BorraTodasOcuurrencias(val:N;VAR L:TipoLista)

(* Solución Iterativa *)

VARIABLES

Ptr,anterior: TipoPuntero

INICIO

Ptr:=L

anterior:=NIL

MIENTRAS (Ptr<>NIL) HACER

SI (Ptr^.valor=val) ENTONCES

SI (anterior=NIL) ENTONCES (* Es el nodo 1º *)

L:=L^.sig

LIBERAR(Ptr,TAMAÑO(Reg))

Ptr:=L

ENOTROCASO

anterior^.sig:= Ptr^.sig

LIBERAR(Ptr,TAMAÑO(Reg))

Ptr:= anterior^.sig

FINSI

ENOTROCASO

anterior:=Ptr

Ptr:=Ptr^.sig

FINSI

FINMIENTRAS

FIN

Ejercicio 4 (a)

ALGORITMO LeeVector():Tvector

VARIABLES

v,Ptr:Tvector

i:[0..MAX-1]

val:Z

INICIO

v:=NIL

PARA i:=0 HASTA MAX-1 HACER

 Escribir("v["i,"]=")

 Leer(val)

 SI val<>0 ENTONCES

 SI v=NIL ENTONCES

 ASIGNAR(v,TAMAÑO(TnodoVector))

 Ptr:=v

 ENOTROCASO

 ASIGNAR(Ptr^.sig,TAMAÑO(TnodoVector))

 Ptr:= Ptr^.sig

 FINSI

 Ptr^.Coord:=i

 Ptr^.valor:=val

 Ptr^.sig:=NIL

 FINSI

FINPARA

DEVOLVER v

FIN

Ejercicio 4 (b)

INICIO (* Ej4 *)

v1:=LeeVector()

v2:=LeeVector()

ptr1:=v1

ptr2:=v2

prod:=0

PARA i:=0 HASTA MAX-1 HACER

(* Cálculo del producto escalar *)

SI (Ptr1=NIL) OR (Ptr1^.Coor>i) ENTONCES

va1:=0

ENOTROCASO

va1:=Ptr1^.valor

Ptr1:=Ptr1^.sig

FINSI

SI (Ptr2=NIL) OR (Ptr2^.Coor>i) ENTONCES

va2:=0

ENOTROCASO

va2:=Ptr2^.valor

Ptr2:=Ptr2^.sig

FINSI

prod:=prod+va1*va2

FINPARA

Escribir("El Producto escalar es : ",prod)

FIN (* Ej4 *)

Ejercicio 5 (a)

TIPOS

Tdigito = [0..9]

TNat = PUNTERO A TnodoNat

TnodoNat = REGISTRO

 Dig:Tdigito

 Sig:TNat

FINREGISTRO

ALGORITMO LeeDigito(VAR d:Tdigito; VAR ok:B)

VARIABLES

 car:\$

INICIO

 Leer(car)

 SI (car>='0') AND (car<='9') ENTONCES

 d:= ORD(car) – ORD('0')

 ok:=TRUE

 ENOTROCASO

 ok:=FALSE

 FINSI

FIN

Ejercicio 5 (b)

ALGORITMO LeeNat():Tnat (* Solución Iterativa *)

VARIABLES

ptr,nat:Tnat

d:Tdigito

ok:B

INICIO

LeeDigito(d,ok)

nat:=NIL

SI ok ENTONCES

ASIGNAR(nat,TAMAÑO(TnodoNat))

nat^.Dig:=d

nat^.Sig:=NIL

ptr:=nat

LeeDigito(d,ok)

MIENTRAS ok HACER

ASIGNAR(ptr^.Sig,TAMAÑO(TnodoNat))

ptr:=ptr^.Sig

ptr^.Dig:=d

ptr^.Sig:=NIL

LeeDigito(d,ok)

FINMIENTRAS

FINSI

DEVOLVER nat

FIN

Ejercicio 5 (c)

ALGORITMO Nat2Real(n:Tnat):R (* Solución Iterativa *)

VARIABLES

Ptr:Tnat

Num:R

INICIO

Ptr:=n

Num:=0.0

MIENTRAS Ptr<>NIL HACER

Num:=Num*10+(Ptr^.Dig)

Ptr:=Ptr^.Sig

FINMIENTRAS

DEVOLVER

FIN

INICIO

DEVOLVER Nat2Real (n,0)

FIN

Ejercicio 6

TIPOS

TipoPolinomio= PUNTERO A TnodoPolinomio

TnodoPolinomio = REGISTRO

coeficiente:Z

grado:N

sig:TipoPolinomio

FINREGISTRO

ALGORITMO potencia(x:Z;y:N):Z (* Solución Iterativa *)

VARIABLES

i:N

pot:Z

INICIO

pot:=1

PARA i:=1 HASTA y HACER

pot:=pot*x

FINPARA

DEVOLVER pot

FIN

ALGORITMO potencia(x:Z;y:N):Z (* Solución Recursiva *)

INICIO

SI y <1 ENTONCES

DEVOLVER 1

ENOTROCASO

DEVOLVER x*potencia(x,y-1)

FINSI

FIN

Ejercicio 6.a

ALGORITMO Evaluar(P:TipoPolinomio;valor:Z):Z

(* Solución Iterativa *)

VARIABLES

val:Z

ptr:TipoPolinomio

INICIO

ptr:=P

val:=0

MIENTRAS ptr<>NIL HACER

 val:=val + ptr^.coeficiente*potencia(valor,ptr^.grado)

 ptr:=ptr^.sig

FINMIENTRAS

DEVOLVER val

FIN

Ejercicio 6.b

ALGORITMO Obtener(P:TipoPolinomio;i:N):Z
(* Solución Iterativa *)

VARIABLES

val:Z

ptr:TipoPolinomio

INICIO

ptr:=P

val:=0

MIENTRAS (ptr<>NIL) AND (ptr^.grado<=i) HACER

SI ptr^.grado=i ENTONCES

 val:=ptr^.coeficiente

FINSI

ptr:=ptr^.sig

FINMIENTRAS

DEVOLVER val

FIN

Ejercicio 7 (a)

ALGORITMO Inserta(x:Z; VAR cjto:Conjunto)
(* Solución Iterativa *)

VARIABLES

Ptr:Conjunto

INICIO

Ptr:=cjto

MIENTRAS (Ptr<>NIL) AND (Ptr^.elem<>x) HACER
(* ¿Pertenece x a cjto ? *)

Ptr:=Ptr^.sig

FINMIENTRAS

SI Ptr=NIL ENTONCES (* x no estaba en cjto *)

ASIGNAR(Ptr,TAMAÑO(Elemento))

Ptr^.elem:=x

Ptr^.sig:=cjto

cjto:=Ptr

FINSI

FIN

Ejercicio 7 (b)

ALGORITMO Union(c1,c2:Conjunto):Conjunto

(* Solución Iterativa *)

VARIABLES

Cjto,ptr:Conjunto

INICIO

Cjto:=NIL

ptr:=c1

MIENTRAS ptr<>NIL HACER (* Copio c1 en Cjto *)

 Insertar(ptr^.elem,Cjto)

 ptr:=ptr^.sig

FINMIENTRAS

ptr:=c2

MIENTRAS ptr<>NIL HACER (* Inserto los elementos de c2
en Cjto *)

 Insertar(ptr^.elem,ptrCjto)

 ptr:=ptr^.sig

FINMIENTRAS

DEVOLVER Cjto

FIN

Ejercicio 8

TIPOS

Tlista = PUNTERO A Tnodo

Tnodo = REGISTRO

Num:N

Sig:Tlista

FINREGISTRO

ALGORITMO LeeLista():Tlista

VARIABLES

L:Tlista

n:N

INICIO

Lee(n)

SI n=0 ENTONCES

DEVOLVER NIL

ENOTROCASO

ASIGNAR(L,TAMAÑO(Tnodo))

L^.Num:=n

L^.Sig:=LeeLista()

DEVOLVER L

FINSI

FIN

Ejercicios 8.1-8.2

ALGORITMO PintaLista(L:Tlista)

INICIO

SI L<> NIL ENTONCES

 Escribir(L^.Num) (* Orden Natural *)

 PintaLista(L^.Sig)

 Escribir(L^.Num) (* Orden Inverso *)

FINSI

FIN

ALGORITMO MayorLista(L:Tlista):N

VARIABLES

 MayorResto:N

INICIO

SI L= NIL ENTONCES

 DEVOLVER 0

ENOTROCASO

 MayorResto:=MayorLista(L^.Sig)

 SI L^.Num> MayorResto ENTONCES

 DEVOLVER L^.Num

 ENOTROCASO

 DEVOLVER MayorResto

 FINSI

FINSI

FIN

Ejercicios 8.3-8.4

ALGORITMO SumaLista(L:Tlista):N

INICIO

SI L= NIL ENTONCES

DEVOLVER 0

ENOTROCASO

DEVOLVER L^.Num+SumaLista(L^.Sig)

FINSI

FIN

ALGORITMO LongitudLista (L:Tlista):N

INICIO

SI L= NIL ENTONCES

DEVOLVER 0

ENOTROCASO

DEVOLVER 1+ LongitudLista (L^.Sig)

FINSI

FIN

<http://www.lcc.uma.es/~afdez/apuntes/>

Visita la página y dinos lo que piensas (si falta algo por ejemplo)