Ejemplo de programas de Listas

15. Programa que permite realizar diferentes operaciones con una lista dinámica simplemente enlazada de números enteros mediante un menú de opciones.

```
program gestiona_pila;
type puntero=^elemento;
     elemento=record
         dato:integer;
          ant:puntero;
         end;
var  primero,p :puntero;
     max_long  :integer;
     opcion    :integer;
     valor     :integer;
procedure crea_pila(var prim:puntero);
     begin
     prim:=nil
     end;
function longitud(prim:puntero):integer;
     var  contador :integer;
                p :puntero;
     begin
     contador:=0;
     p:=prim;
     while p<>nil do
           begin
           contador:=contador+1;
           p:=p^.ant
           end;
     longitud:=contador
     end;
procedure insertar(valor:integer;var prim:puntero;l:integer);
     var  p :puntero;
     begin
     if longitud(prim)<>l
          then begin
               new(p);
               p^.dato:=valor;
               p^.ant:=prim;
               prim:=p
               end
          else writeln('pila llena')
     end;
procedure borrar(var prim:puntero);
     var  p :puntero;
     begin
     if prim<>nil
          then begin
               p:=prim;
               prim:=p^.ant;
               dispose(p)
               end
          else writeln('pila vacia')
     end;
procedure visualiza(prim:puntero);
     begin
     p:=prim;
     while p<>nil do
           begin
           writeln(p^.dato);
           p:=p^.ant
           end
     end;
{ *******  Programa principal  ******* }
```

```
begin
opcion:=1;
writeln('Tama¤o de la pila : '); readln(max_long);
while opcion<>0 do
     begin
     writeln('1=crear, 2=insertar, 3=borrar, 4=consultar');
     writeln('5=visualizar pila, 6=longitud de la pila, 0=fin');
     readln(opcion);
     case opcion of
          1: crea_pila(primero);
          2: begin
             writeln('intro valor del elemento'); readln(valor);
             insertar(valor,primero,max_long)
             end;
          3: borrar(primero);
          4: begin
             if primero<>nil
                 then writeln('El valor del ultimo elemento es ',primero^.dato)
                 else writeln('Pila vacia.')
             end;
          5: visualiza(primero);
          6: writeln('El tama¤o de la pila es ',longitud(primero))
          end
     end
end.
```

16. Programa que multiplica dos polinomios cuyos coeficientes están almacenados en listas doblemente enlazadas. El polinomio resultante también es almacenado en otra lista doblemente enlazada.

```
(*********************************************************)
(* Programa producto_polinomios                          *)
(* Multiplica dos polinomios almacenados en listas dobles.  *)
(* El producto tambien es almacenado en otra lista       *)
(* Division de Informatica Ind. ETSI Industriales. UPM   *)
(*********************************************************)
program producto_polinomios;
type puntero = ^dato;
       dato  = record
       val       : real;
       sig, pred : puntero;
       end;
var poli1, poli2, t : puntero; grado1, grado2, g : integer;
procedure leer(var r:puntero; var g:integer);
      var   k : integer; aux : real; u : puntero;
      begin
      write('Grado del polinomio : ');
      readln(g);
      new(r);
      u:=r;
      u^.pred:=nil;
      for k:=0 to g do
            begin
            write('Coeficiente de grado ',k:3,'  ');
            readln(aux);
            u^.val:=aux;
            if k<g then
            begin
                  new(u^.sig);
                  u^.sig^.pred:=u;
                  u:=u^.sig
                  end
             else
                  u^.sig:=nil
           end
     end;
procedure escribir(r:puntero);
     var s:puntero;
```

```
     begin
     s:=r;
     while s<>nil do
            begin
            write(s^.val:7:2,'  ');
            s:=s^.sig
            end;
     writeln
     end;
procedure producto(r1,r2:puntero; g1,g2:integer;
                        var r:puntero; var g:integer);
      var   k : integer; aux : real; s, t, u, auxp : puntero;
      begin
      g:=g1+g2;
      s:=r1;
      t:=r2;
      new(r);
      u:=r;
      u^.pred:=nil;
      u^.val:=s^.val*t^.val;
      for k:=1 to g do
            begin
            if s^.sig=nil
            then  begin
                  auxp:=s;
                  s:=t^.sig;
                  t:=auxp
                  end
            else  begin
                  auxp:=t;
                  t:=s^.sig;
                  s:=auxp
                  end;
          aux:=0.0;
          while (s^.sig<>nil) and (t^.pred<>nil) do
                  begin
                  aux:=aux+s^.val*t^.val;
                  s:=s^.sig;
                  t:=t^.pred
                  end;
          new(u^.sig);
          u^.sig^.pred:=u;
           u:=u^.sig;
          u^.val:=aux+s^.val*t^.val
          end;
     u^.sig:=nil
     end;
{ ******  Programa principal  ****** }
begin
leer(poli1,grado1);
leer(poli2,grado2);
producto(poli1,poli2,grado1,grado2,t,g);
escribir(t)
end.
```