

Introducción a la Programación I – Práctico 5 (Variables)

```
program elp5;
  {Intercambia los valos de dos variables de tipo Integer}

  procedure cambiavalores (var primera:Integer; var segunda:Integer);

  var Aux:Integer;

  Begin
    Aux:= Primera;
    Primera:= Segunda;
    Segunda:= Aux;
  End;

  var enteral, Entera2:Integer;

  begin
    Enteral:= 2;
    Entera2:= 5;
    Writeln('Antes de moverse las variables:');
    Writeln('Variable 1:', Enteral);
    Writeln('Variable 2:', Entera2);
    cambiavalores (Enteral, Entera2);
    Writeln('Despues de moverse:');
    Writeln('Variable 1: ', Enteral);
    Writeln('Variable 2: ', Entera2);
  end.

  /*****/
```

```
program e2p5;
  {Suma los primeros N numeros naturales, N es definido por teclado}

procedure sumanumeros (vueltas:Integer;var total:Integer);
  {Proceso que realiza la suma requerida}

Var Siguiente, Actual:Integer;

Begin
  Actual:= 0;
  While Actual <> Vueltas do
    Begin
      Siguiente:=Actual + 1;
      Total:= Total + Siguiente;
      Actual:=Actual + 1 ;
    End;
End;

var Suma, Cantnumeros:Integer;

begin
  Cantnumeros:= 6;
  Writeln('Escribe la cantidad N numeros que deseas sumar:');
  Readln(cantnumeros);
  Suma:= 0;
  sumanumeros(Cantnumeros, Suma);
  Writeln('La suma de los primeros ', Cantnumeros,
  'numeros naturales es igual a: ', Suma);
end.

/*****/
```

```
Program e3p5;
{determina cuanto recibe cada socio, si se distribuye equitativamente
el total del dinero}
uses estructu;
{.....}
procedure sumarcheques (origen:fila; var destino:integer);
{suma todos los valores de la fila origen, y da el resultado en destino}
var aux:pila;
begin
  inicpila(aux, '');
  if not filavacia (origen) then
    apilar(aux, extraer(origen));
  destino:=(tope(aux));
  while not filavacia(origen) do
    begin
      apilar(aux, extraer(origen));
      destino:=(destino+(tope(aux)));
    end;
  end;
{.....}
procedure sumarsocios (origen:fila; var destino:integer);
{suma la cantidad de elementos de la fila origen}
var aux1:pila;
begin
  inicpila(aux1, '');
  destino:=0;
  while not filavacia(origen) do
    begin
      apilar(aux1, extraer(origen));
      destino:=(destino+1);
    end;
  end;
{.....}
var cheques, socios:fila;
  sumacheques, sumasocios, dinero:integer;
begin
  readfila(socios);
  readfila(cheques);
  sumacheques:=0;
  sumasocios:=0;
  sumarcheques(cheques, sumacheques);
  sumarsocios(socios, sumasocios);
  dinero:=(sumacheques div sumasocios);
  writeln('dinero que le corresponde a cada socio:');
  writeln(dinero);
end.

/*****/
```

```
program e4p5;
  {dada 2 fechas, dice, si la primera es mas grande que la segunda}
var A1,M1,D1,A2,M2,D2:integer;
  resultado:boolean;
begin
  readln(A1);
  readln(M1);
  readln(D1);
  readln(A2);
  readln(M2);
  readln(D2);
  if (A1) < (A2) then
    resultado:=True;
  if (A1) > (A2) then
    resultado:=False;
  if ((A1) = (A2)) and
    ((M1) < (M2)) then
    resultado:=True;
  if ((A1) = (A2)) and
    ((M1) > (M2)) then
    resultado:=False;
  if ((A1) = (A2)) and
    ((M1) = (M2)) and
    ((D1) < (D2)) then
    resultado:=True;
  if ((A1) = (A2)) and
    ((M1) = (M2)) and
    ((D1) > (D2)) then
    resultado:=False;
  if ((A1) = (A2)) and
    ((M1) = (M2)) and
    ((D1) = (D2)) then
    resultado:=False;
  writeln(resultado);
end.
```

```
/*  
*****  
*/
```

```
program e5p5;  
{da como resultado la cantidad de dias entre 2 fechas}  
{.....}  
Procedure resta(inicial1,inicial2:integer;var destino:integer);  
{resta inicial1 a inicial2, y deja el resultado en destino}  
begin  
  destino:=(inicial2 - inicial1);  
end;  
{.....}  
var A1,M1,D1,A2,M2,D2,cantdias,dias,meses,anios:integer;  
begin  
  readln(A1);  
  readln(M1);  
  readln(D1);  
  readln(A2);  
  readln(M2);  
  readln(D2);  
  anios:=0;  
  resta(A1,A2,anios);  
  anios:=(anios*360);  
  meses:=0;  
  resta(M1,M2,meses);  
  meses:=(meses*30);  
  dias:=0;  
  resta(D1,D2,dias);  
  cantdias:=(anios + meses + dias);  
  writeln(cantdias);  
end.
```

```
/*****/
```

```
Program e6p5;
{nos dice en que posicion de la fila se encuentra un elemento que
buscamos}
uses estructu;
{.....}
procedure encontrarelemento (var
ubicacion:integer;elemento1:integer;origen:fila);
{busca en la fila origen la posicion de elemento1, y la devuelve en
ubicacion}
var descarte:fila;
begin
  ubicacion:=1;
  inicfila(descarte, '');
  while (not filavacia (origen)) and
    (primero(origen) <> (elemento1)) do
    begin
      ubicacion:=ubicacion+1;
      agregar(descarte, extraer(origen));
    end;
  if filavacia (origen) then
    ubicacion:=0;
end;
{.....}
var elemento,posicion:integer;original:fila;
begin
  elemento:=0;
  readfila(original);
  write('Ingrese el Elemento a Buscar ');
  read(elemento);
  posicion:=0;
  if not filavacia (original) then
    encontrarelemento(posicion,elemento,original);
  if (posicion)=0 then
    writeln('Elemento no Encontrado')
  else
    writeln('Posicion del Elemento ', posicion);
end.

/*****/
```

```
Program e7p5;
{dado un numero, elimina de la pila, el elemento que se encuentra en esa
posicion}
uses estructu;

procedure eliminar(var origen:pila; numero:integer);
{elimina de la pila origen el elemento que se encuentra en la posicion
"numero"}
var descarte:pila; aux:pila;
begin
  inicpila(descarte, '');
  inicpila(aux, '');
  while numero <> 0 do
    begin
      apilar(aux, desapilar(origen));
      numero:= numero-1;
    end;
  if (not pilavacia (aux)) and
    (numero=0) then
    apilar(descarte, desapilar(aux));
  while not pilavacia (aux) do
    apilar(origen, desapilar(aux));
end;

var posicion:integer; secuencia:pila;
begin
  posicion:=0;
  writeln('Ingrese los Elementos de la Secuencia');
  readpila(secuencia);
  writeln('Ingrese la Posicion del Elemto que Desea Eliminar  ');
  read(posicion);
  eliminar(secuencia, posicion);
  writepila(secuencia);
end.

/*****/
```

```
program e8p5;

uses Estructu;

procedure borrarelemento (Pos:Integer; var Sec:Pila);

var Aux, Basura:Pila;
    contador:Integer;

Begin
  Inicpila (Aux, '');
  Inicpila (Basura, '');
  Contador:= 0;
  While (not pilavacia (Sec)) and
    (Contador <> Pos) do
    Begin
      Apilar (Aux, desapilar (Sec));
      Contador:= Contador + 1;
    End;
  If not pilavacia (Aux) Then
    Apilar (Basura, desapilar (Aux));
  While not pilavacia (Aux) do
    Apilar (Sec, desapilar (Aux));
End;

procedure ordenarpila (var pos:Pila;var aux2:pila);

var temp:Pila;

Begin
  Inicpila (temp, '');
  Inicpila (Aux2, '');
  If not pilavacia (pos) then
    Apilar (Aux2, desapilar (pos));
  While not pilavacia (pos) do
    If Tope (pos) > Tope (Aux2) Then
      Begin
        Apilar (temp, desapilar (Aux2));
        Apilar (Aux2, desapilar (pos));
      End
    Else
      Apilar (Aux2, desapilar (Pos));
    While not pilavacia (temp) do
      Apilar (pos, desapilar (temp));
End;

var ORIGINAL, POSICIONES, Auxiliar:Pila;

    posicion:Integer;
```

Begin

Readpila (ORIGINAL);

Readpila (POSICIONES);

While not pilavacia (Posiciones) **do**

 ordenarpila (Posiciones, Auxiliar);

While not pilavacia (Auxiliar) **do**

 Apilar (Posiciones, desapilar (Auxiliar));

While not pilavacia (Posiciones) **do**

Begin

 posicion:= Desapilar (Posiciones);

 borrarelemento (posicion, Original);

End;

Writepila (Original);

End.