

Trabajo Práctico N° 4.1: Vectores (Parte 1).

Ejercicio 1.

```
program TP4_E1;
{$codepage UTF8}
uses crt;
type
  vnums=array[1..10] of integer;
var
  numeros: vnums;
  i: integer;
begin
  for i:=1 to 10 do
    numeros[i]:= i;
  for i:= 2 to 10 do
    numeros[i]:=numeros[i]+numeros[i-1]
  end.
end.
```

(a) *¿Qué valores toma la variable numeros al finalizar el primer bloque for?*

Los valores que toma la variable numeros al finalizar el primer bloque for son 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

(b) *Al terminar el programa, ¿con qué valores finaliza la variable numeros?*

Al terminar el programa, la variable numeros finaliza con los valores 1, 3, 6, 10, 15, 21, 28, 36, 45, 55.

(c) *Si se desea cambiar la línea 11 por la sentencia: for i:=1 to 9 do, ¿cómo debe modificarse el código para que la variable números contenga los mismos valores que en (1.b)?*

```
program TP4_E1c;
{$codepage UTF8}
uses crt;
type
  vnums=array[1..10] of integer;
var
  numeros: vnums;
  i, j: integer;
begin
  for i:= 1 to 10 do
    numeros[i]:= i;
  for i:= 1 to 9 do
    begin
      j:=i+1;
      numeros[j]:=numeros[j]+numeros[i];
      writeln(numeros[j]);
    end;
  end.
end.
```

(d) ¿Qué valores están contenidos en la variable `numeros` si las líneas 11 y 12 se reemplazan por `for i:= 1 to 9 do numeros[i+1]:=numeros[i];`?

```
program TP4_E1d;
{$codepage UTF8}
uses crt;
type
  vnums=array[1..10] of integer;
var
  numeros: vnums;
  i, j: integer;
begin
  for i:= 1 to 10 do
    numeros[i]:= i;
  for i:= 1 to 9 do
    begin
      numeros[i+1]:=numeros[i];
      writeln(numeros[i]);
    end;
end.
```

Ejercicio 2.

Dado el siguiente programa, completar las líneas indicadas, considerando que:

- El módulo cargarVector debe leer números reales y almacenarlos en el vector que se pasa como parámetro. Al finalizar, debe retornar el vector y su dimensión lógica. La lectura finaliza cuando se ingresa el valor 0 (que no debe procesarse) o cuando el vector está completo.
- El módulo modificarVectorySumar debe devolver el vector con todos sus elementos incrementados con el valor n y también debe devolver la suma de todos los elementos del vector.

```

program TP4_E2;
{$codepage UTF8}
uses crt;
const
    cant_datos=150;
type
    vdatos=array[1..cant_datos] of real;
procedure cargarVector(var v: vdatos; var dimL: integer);
var
    num_real: int16;
begin
    textcolor(green); write('Introducir número real: ');
    textcolor(yellow); readln(num_real);
    while ((num_real<>0) and (dimL<=cant_datos)) do
    begin
        dimL:=dimL+1;
        v[dimL]:=num_real;
        textcolor(green); write('Introducir número real: ');
        textcolor(yellow); readln(num_real);
    end;
end;
procedure modificarVectorySumar(var v: vdatos; dimL: integer; n: real; var suma: real);
var
    i: int16;
begin
    for i:= 1 to dimL do
    begin
        v[i]:=v[i]+n;
        suma:=suma+v[i];
    end;
end;
var
    datos: vdatos;
    i, dim: integer;
    num, suma: real;
begin
    dim:=0; suma:=0;
    cargarVector(datos,dim);
    textcolor(green); write('Introducir valor a sumar: ');
    textcolor(yellow); readln(num);
    modificarVectorySumar(datos,dim,num,suma);
    textcolor(green); write('La suma de los valores es '); textcolor(red); writeln(suma:0:2);
    textcolor(green); write('Se procesaron '); textcolor(red); write(dim); textcolor(green);
    write(' números');
end.

```

Ejercicio 3.

Se dispone de un vector con números enteros, de dimensión física $dimF$ y dimensión lógica $dimL$.

(a) Realizar un módulo que imprima el vector desde la primera posición hasta la última.

```
procedure imprimir_1adimL(vector: t_vector; dimL: int16);
var
  i: int16;
begin
  for i:= 1 to dimL do
    writeln(vector[i]);
  end;
```

(b) Realizar un módulo que imprima el vector desde la última posición hasta la primera.

```
procedure imprimir_dimLa1(vector: t_vector; dimL: int16);
var
  i: int16;
begin
  for i:= dimL downto 1 do
    writeln(vector[i]);
  end;
```

(c) Realizar un módulo que imprima el vector desde la mitad ($dimL \text{ DIV } 2$) hacia la primera posición, y desde la mitad más uno hacia la última posición.

```
procedure imprimir_dimLdiv2(vector: t_vector; dimL: int16);
var
  i, dimLdiv2, dimLdiv2mas1: int16;
begin
  dimLdiv2:=dimL div 2; dimLdiv2mas1:=dimLdiv2+1;
  for i:= dimLdiv2 downto 1 do
    writeln(vector[i]);
  for i:= dimLdiv2mas1 to dimL do
    writeln(vector[i]);
  end;
```

(d) Realizar un módulo que reciba el vector, una posición X y otra posición Y , e imprima el vector desde la posición X hasta la Y . Asumir que tanto X como Y son menores o igual a la dimensión lógica. Y considerar que, dependiendo de los valores de X e Y , podría ser necesario recorrer hacia adelante o hacia atrás.

```
procedure imprimir_general(vector: t_vector);
var
  i, numX, numY: int16;
begin
  textcolor(green); write('Introducir número entero X: ');
  textcolor(yellow); readln(numX);
```

```

textcolor(green); write('Introducir número entero Y: ');
textcolor(yellow); readln(numY);
if (numX<=numY) then
begin
    for i:= numX to numY do
        writeln(vector[i]);
    end
else
    for i:= numX downto numY do
        writeln(vector[i]);
    end;
end;

```

(e) Utilizando el módulo implementado en el inciso anterior, volver a realizar los incisos a, b y c.

```

program TP4_E3;
{$codepage UTF8}
uses crt;
type
    t_vector=array of int16;
procedure crear_vector(var vector: t_vector; dimF: int16);
begin
    setLength(vector,dimF);
end;
procedure rellenar_vector(var vector: t_vector; limite_random, dimL: int16);
var
    i: int16;
begin
    for i:= 1 to dimL do
        vector[i]:=random(limite_random);
    end;
end;
procedure imprimir_1adimL(vector: t_vector; dimL: int16);
var
    i: int16;
begin
    for i:= 1 to dimL do
        writeln(vector[i]);
    end;
end;
procedure imprimir_dimLa1(vector: t_vector; dimL: int16);
var
    i: int16;
begin
    for i:= dimL downto 1 do
        writeln(vector[i]);
    end;
end;
procedure imprimir_dimLdiv2(vector: t_vector; dimL: int16);
var
    i, dimLdiv2, dimLdiv2mas1: int16;
begin
    dimLdiv2:=dimL div 2; dimLdiv2mas1:=dimLdiv2+1;
    for i:= dimLdiv2 downto 1 do
        writeln(vector[i]);
    end;
    for i:= dimLdiv2mas1 to dimL do
        writeln(vector[i]);
    end;
end;
procedure imprimir_general(vector: t_vector);
var
    i, numX, numY: int16;
begin
    textcolor(green); write('Introducir número entero X: ');
    textcolor(yellow); readln(numX);
    textcolor(green); write('Introducir número entero Y: ');

```

```
textcolor(yellow); readln(numY);
if (numX<=numY) then
begin
  for i:= numX to numY do
    writeln(vector[i]);
  end
else
  for i:= numX downto numY do
    writeln(vector[i]);
end;
var
  vector: t_vector;
  limite_random, dimF, dimL: int16;
begin
  randomize;
  textcolor(green); write('Introducir número entero como límite superior de una distribución
aleatoria de la cual se extraerán elementos para el vector: ');
  textcolor(yellow); readln(limite_random);
  textcolor(green); write('Introducir número entero como dimensión física del vector: ');
  textcolor(yellow); readln(dimF);
  textcolor(green); write('Introducir número entero como dimensión lógica del vector: ');
  textcolor(yellow); readln(dimL);
  crear_vector(vector,dimF);
  rellenar_vector(vector,limite_random,dimL);
  imprimir_1adimL(vector,dimL);
  imprimir_dimLa1(vector,dimL);
  imprimir_dimLdiv2(vector,dimL);
  imprimir_general(vector);
end.
```

Ejercicio 4.

Se dispone de un vector con 100 números enteros. Implementar los siguientes módulos:

(a) posicion: dado un número X y el vector de números, retorna la posición del número X en dicho vector, o el valor -1 en caso de no encontrarse.

```
function posicion(vector: t_vector; dimL, numX: int16): int16;
var
  pos: int16;
begin
  pos:=0;
  while ((pos<=dimL) and (vector[pos]<>numX)) do
    pos:=pos+1;
  if (pos>dimL) then
    pos:=-1;
  posicion:=pos;
end;
```

(b) intercambio: recibe dos valores x e y (entre 1 y 100) y el vector de números, y retorna el mismo vector, donde se intercambiaron los valores de las posiciones x e y .

```
procedure intercambio(var vector: t_vector; numX, numY: int16);
var
  num_aux: int16;
begin
  num_aux:=vector[numX];
  vector[numX]:=vector[numY];
  vector[numY]:=num_aux;
end;
```

(c) sumaVector: retorna la suma de todos los elementos del vector.

```
function sumaVector(vector: t_vector; dimL: int16): int16;
var
  i, suma: int16;
begin
  suma:=0;
  for i:= 1 to dimL do
    suma:=suma+vector[i];
  sumaVector:=suma;
end;
```

(d) promedio: devuelve el valor promedio de los elementos del vector.

```
function promedio(vector: t_vector; dimL: int16): real;
begin
  promedio:=sumaVector(vector,dimL)/dimL;
end;
```

(e) *elementoMaximo*: retorna la posición del mayor elemento del vector.

```
function elementoMaximo(vector: t_vector; dimL: int16): int16;
var
  i, ele_max, pos_max: int16;
begin
  ele_max:=low(int16); pos_max:=low(int16);
  for i:= 1 to dimL do
    begin
      if (vector[i]>ele_max) then
        begin
          ele_max:=vector[i];
          pos_max:=i;
        end;
      end;
    elementoMaximo:=pos_max;
  end;
```

(f) *elementoMinimo*: retorna la posición del menor elemento del vector.

```
function elementoMinimo(vector: t_vector; dimL: int16): int16;
var
  i, ele_min, pos_min: int16;
begin
  ele_min:=high(int16); pos_min:=high(int16);
  for i:= 1 to dimL do
    begin
      if (vector[i]<ele_min) then
        begin
          ele_min:=vector[i];
          pos_min:=i;
        end;
      end;
    elementoMinimo:=pos_min;
  end;
```

```
program TP4_E4;
{$codepage UTF8}
uses crt;
type
  t_vector=array of int16;
procedure crear_vector(var vector: t_vector; dimF: int16);
begin
  setLength(vector,dimF);
end;
procedure rellenar_vector(var vector: t_vector; dimL: int16);
var
  i: int16;
begin
  for i:= 1 to dimL do
    vector[i]:=random(maxint);
  end;
function posicion(vector: t_vector; dimL, numX: int16): int16;
var
  pos: int16;
begin
  pos:=0;
  while ((pos<=dimL) and (vector[pos]<>numX)) do
    pos:=pos+1;
  if (pos>dimL) then
```



```

    pos:=-1;
    posicion:=pos;
end;
procedure intercambio(var vector: t_vector; numX, numY: int16);
var
    num_aux: int16;
begin
    num_aux:=vector[numX];
    vector[numX]:=vector[numY];
    vector[numY]:=num_aux;
end;
function sumaVector(vector: t_vector; dimL: int16): int16;
var
    i, suma: int16;
begin
    suma:=0;
    for i:= 1 to dimL do
        suma:=suma+vector[i];
    sumaVector:=suma;
end;
function promedio(vector: t_vector; dimL: int16): real;
begin
    promedio:=sumaVector(vector,dimL)/dimL;
end;
function elementoMaximo(vector: t_vector; dimL: int16): int16;
var
    i, ele_max, pos_max: int16;
begin
    ele_max:=low(int16); pos_max:=low(int16);
    for i:= 1 to dimL do
        begin
            if (vector[i]>ele_max) then
                begin
                    ele_max:=vector[i];
                    pos_max:=i;
                end;
        end;
    elementoMaximo:=pos_max;
end;
function elementoMinimo(vector: t_vector; dimL: int16): int16;
var
    i, ele_min, pos_min: int16;
begin
    ele_min:=high(int16); pos_min:=high(int16);
    for i:= 1 to dimL do
        begin
            if (vector[i]<ele_min) then
                begin
                    ele_min:=vector[i];
                    pos_min:=i;
                end;
        end;
    elementoMinimo:=pos_min;
end;
var
    vector: t_vector;
    dimF, dimL, numX, numY: int16;
begin
    textcolor(green); write('Introducir número entero como dimensión física del vector: ');
    textcolor(yellow); readln(dimF);
    textcolor(green); write('Introducir número entero como dimensión lógica del vector: ');
    textcolor(yellow); readln(dimL);
    crear_vector(vector,dimF);
    rellenar_vector(vector,dimL);
    textcolor(green); write('Introducir número entero X para buscar su posición (si existe) en
el vector: ');

```

```
    textcolor(yellow); readln(numX);
    textcolor(green); write('La posición del número '); textcolor(red); write(numX);
textcolor(green); write(' en el vector es '); textcolor(red);
writeln(posicion(vector,dimL,numX));
    textcolor(green); write('Introducir número entero X (entre 1 y 100): ');
    textcolor(yellow); readln(numX);
    textcolor(green); write('Introducir número entero Y (entre 1 y 100): ');
    textcolor(yellow); readln(numY);
    textcolor(green); write('Pre-intercambio, en las posiciones '); textcolor(red);
write(numX); textcolor(green); write(', '); textcolor(red); write(numY); textcolor(green);
write(', se tienen los valores '); textcolor(red); write(vector[numX]); textcolor(green);
write(' y '); textcolor(red); write(vector[numY]); textcolor(green); writeln(',
respectivamente');
    intercambio(vector,numX,numY);
    textcolor(green); write('Post-intercambio, en las posiciones '); textcolor(red);
write(numX); textcolor(green); write(', '); textcolor(red); write(numY); textcolor(green);
write(', se tienen los valores '); textcolor(red); write(vector[numX]); textcolor(green);
write(' y '); textcolor(red); write(vector[numY]); textcolor(green); writeln(',
respectivamente');
    textcolor(green); write('La suma de todos los elementos del vector es '); textcolor(red);
writeln(sumaVector(vector,dimL));
    textcolor(green); write('El valor promedio de los elementos del vector es ');
textcolor(red); writeln(promedio(vector,dimL):0:2);
    textcolor(green); write('La posición del mayor elemento del vector es '); textcolor(red);
writeln(elementoMaximo(vector,dimL));
    textcolor(green); write('La posición del menor elemento del vector es '); textcolor(red);
write(elementoMinimo(vector,dimL));
end.
```

Ejercicio 5.

Utilizando los módulos implementados en el ejercicio 4, realizar un programa que lea números enteros desde teclado (a lo sumo, 100) y los almacene en un vector. La carga finaliza al leer el número 0. Al finalizar la carga, se debe intercambiar la posición del mayor elemento por la del menor elemento e informar la operación realizada de la siguiente manera: “El elemento máximo ... que se encontraba en la posición ... fue intercambiado con el elemento mínimo ... que se encontraba en la posición ...”.

```

program TP4_E5;
{$codepage UTF8}
uses crt;
const
    dimF=100; vector_salida=0;
type
    t_vector=array[1..dimF] of int16;
procedure rellenar_vector(var vector: t_vector; var dimL: int16; dimF: int16);
var
    num: int16;
begin
    textcolor(green); write('Introducir número entero para la posición ', dimL+1, ' del
vector: ');
    textcolor(yellow); readln(num);
    while ((num<>vector_salida) and (dimL<dimF)) do
    begin
        dimL:=dimL+1;
        vector[dimL]:=num;
        textcolor(green); write('Introducir número entero para la posición ', dimL+1, ' del
vector: ');
        textcolor(yellow); readln(num);
    end;
end;
procedure intercambio(var vector: t_vector; pos_max, pos_min: int16);
var
    num_aux: int16;
begin
    num_aux:=vector[pos_max];
    vector[pos_max]:=vector[pos_min];
    vector[pos_min]:=num_aux;
end;
procedure elementoMaximo(vector: t_vector; dimL: int16; var ele_max, pos_max: int16);
var
    i: int16;
begin
    for i:= 1 to dimL do
    begin
        if (vector[i]>ele_max) then
        begin
            ele_max:=vector[i];
            pos_max:=i;
        end;
    end;
end;
procedure elementoMinimo(vector: t_vector; dimL: int16; var ele_min, pos_min: int16);
var
    i: int16;
begin
    for i:= 1 to dimL do
    begin
        if (vector[i]<ele_min) then
        begin
            ele_min:=vector[i];

```

```
        pos_min:=i;
    end;
end;
end;
var
    vector: t_vector;
    dimL, ele_max, ele_min, pos_max, pos_min: int16;
begin
    dimL:=0; ele_max:=low(int16); ele_min:=high(int16); pos_max:=0; pos_min:=0;
    rellenar_vector(vector,dimL,dimF);
    elementoMaximo(vector,dimL,ele_max,pos_max);
    elementoMinimo(vector,dimL,ele_min,pos_min);
    intercambio(vector,pos_max,pos_min);
    textcolor(green); write('El elemento máximo '); textcolor(red); write(ele_max);
textcolor(green); write(', que se encontraba en la posición '); textcolor(red);
write(pos_max); textcolor(green); write(', fue intercambiado con el elemento mínimo ');
textcolor(red); write(ele_min); textcolor(green); write(', que se encontraba en la posición
'); textcolor(red); write(pos_min);
end.
```

Ejercicio 6.

Dado que en la solución anterior se recorre dos veces el vector (una para calcular el elemento máximo y otra para el mínimo), implementar un único módulo que recorra una única vez el vector y devuelva ambas posiciones.

```

program TP4_E6;
{$codepage UTF8}
uses crt;
const
    dimF=100; vector_salida=0;
type
    t_vector=array[1..dimF] of int16;
procedure rellenar_vector(var vector: t_vector; var dimL: int16; dimF: int16);
var
    num: int16;
begin
    textcolor(green); write('Introducir número entero para la posición ', dimL+1, ' del
vector: ');
    textcolor(yellow); readln(num);
    while ((num<>vector_salida) and (dimL<dimF)) do
        begin
            dimL:=dimL+1;
            vector[dimL]:=num;
            textcolor(green); write('Introducir número entero para la posición ', dimL+1, ' del
vector: ');
            textcolor(yellow); readln(num);
        end;
    end;
procedure intercambio(var vector: t_vector; pos_max, pos_min: int16);
var
    num_aux: int16;
begin
    num_aux:=vector[pos_max];
    vector[pos_max]:=vector[pos_min];
    vector[pos_min]:=num_aux;
end;
procedure elementosMaximoYMinimo(vector: t_vector; dimL: int16; var ele_max, ele_min,
pos_max, pos_min: int16);
var
    i: int16;
begin
    for i:= 1 to dimL do
        begin
            if (vector[i]>ele_max) then
                begin
                    ele_max:=vector[i];
                    pos_max:=i;
                end;
            if (vector[i]<ele_min) then
                begin
                    ele_min:=vector[i];
                    pos_min:=i;
                end;
        end;
    end;
end;
var
    vector: t_vector;
    dimL, ele_max, ele_min, pos_max, pos_min: int16;
begin
    dimL:=0; ele_max:=low(int16); ele_min:=high(int16); pos_max:=0; pos_min:=0;
    rellenar_vector(vector,dimL,dimF);
    elementosMaximoYMinimo(vector,dimL,ele_max,ele_min,pos_max,pos_min);

```

```
intercambio(vector,pos_max,pos_min);  
    textcolor(green); write('El elemento máximo '); textcolor(red); write(ele_max);  
textcolor(green); write(', que se encontraba en la posición '); textcolor(red);  
write(pos_max); textcolor(green); write(', fue intercambiado con el elemento mínimo ');  
textcolor(red); write(ele_min); textcolor(green); write(', que se encontraba en la posición  
'; textcolor(red); write(pos_min);  
end.
```

Ejercicio 7.