**Trabajo Práctico N° 4.1:**

**Vectores (Parte 1).**

**Ejercicio 1.**

*Dado el siguiente programa:*

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.

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

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

program TP4\_E1ab;

*{$codepage UTF8}*

uses crt;

type

  vnums=array[1..10] of integer;

var

  numeros: vnums;

  i: integer;

begin

  for i:= 1 to 10 do

  begin

    numeros[i]:=i;

    if (i<10) then

      write(numeros[i],', ')

    else

      writeln(numeros[i])

  end;

  for i:= 2 to 10 do

  begin

    numeros[i]:=numeros[i]+numeros[i-1];

    if (i<10) then

      write(numeros[i],', ')

    else

      writeln(numeros[i])

  end;

end.

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

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: integer;

begin

  for i:= 1 to 10 do

  begin

    numeros[i]:=i;

    if (i<10) then

      write(numeros[i],', ')

    else

      writeln(numeros[i])

  end;

  for i:= 1 to 9 do

  begin

    numeros[i+1]:=numeros[i+1]+numeros[i];

    if (i+1<10) then

      write(numeros[i+1],', ')

    else

      writeln(numeros[i+1])

  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: integer;

begin

  for i:= 1 to 10 do

  begin

    numeros[i]:=i;

    if (i<10) then

      write(numeros[i],', ')

    else

      writeln(numeros[i])

  end;

  for i:= 1 to 9 do

  begin

    numeros[i+1]:=numeros[i];

    if (i<9) then

      write(numeros[i],', ')

    else

      writeln(numeros[i])

  end;

end.

Los valores que 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]* son 1, 1, 1, 1, 1, 1, 1, 1, 1, 1.

**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;

  num\_salida=0;

type

  vdatos=array[1..cant\_datos] of real;

procedure cargarVector(var v: vdatos; var dimL: integer);

var

  num: real;

begin

  num:=num\_salida+random(101)/10;

  while ((num<>num\_salida) and (dimL<cant\_datos)) do

  begin

    dimL:=dimL+1;

    v[dimL]:=num;

    num:=num\_salida+random(101)/10;

  end;

end;

var

  num: real;

begin

  num:=random(101)/10;

  while ((num<>num\_salida) and (dimL<cant\_datos)) do

  begin

    dimL:=dimL+1;

    v[dimL]:=num;

    num:=random(101)/10;

  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;

  dim: integer;

  num, suma: real;

begin

  randomize;

  dim:=0; suma:=0;

  cargarVector(datos,dim);

  num:=1+random(10);

  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.*

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

**(c)** *Realizar un módulo que imprima el vector desde la mitad (dimL DIV 2) hacia la primera posición y desde la mitad más uno hacia la última posición.*

**(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.*

**(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\_numeros=array of int16;

procedure crear\_vector\_numeros(var vector\_numeros: t\_vector\_numeros; dimF: int16);

begin

  setLength(vector\_numeros,dimF);

end;

procedure cargar\_vector\_numeros(var vector\_numeros: t\_vector\_numeros; dimL: int16);

var

  i: int16;

begin

  for i:= 1 to dimL do

    vector\_numeros[i]:=random(high(int16));

end;

procedure imprimir\_1adimL(vector\_numeros: t\_vector\_numeros; dimL: int16);

var

  i: int16;

begin

  for i:= 1 to dimL do

    if (i<dimL) then

      write(vector\_numeros[i],', ')

    else

      writeln(vector\_numeros[i]);

end;

procedure imprimir\_dimLa1(vector\_numeros: t\_vector\_numeros; dimL: int16);

var

  i: int16;

begin

  for i:= dimL downto 1 do

    if (i>1) then

      write(vector\_numeros[i],', ')

    else

      writeln(vector\_numeros[i]);

end;

procedure imprimir\_dimLdiv2(vector\_numeros: t\_vector\_numeros; dimL: int16);

var

  i, dimLdiv2, dimLdiv2mas1: int16;

begin

  dimLdiv2:=dimL div 2; dimLdiv2mas1:=dimLdiv2+1;

  for i:= dimLdiv2 downto 1 do

    if (i>1) then

      write(vector\_numeros[i],', ')

    else

      writeln(vector\_numeros[i]);

  for i:= dimLdiv2mas1 to dimL do

    if (i<dimL) then

      write(vector\_numeros[i],', ')

    else

      writeln(vector\_numeros[i]);

end;

procedure imprimir\_general(vector\_numeros: t\_vector\_numeros; numX, numY: int16);

var

  i: int16;

begin

  if (numX<=numY) then

    for i:= numX to numY do

      if (i<numY) then

        write(vector\_numeros[i],', ')

      else

        writeln(vector\_numeros[i])

  else

    for i:= numX downto numY do

      if (i>numY) then

        write(vector\_numeros[i],', ')

      else

        writeln(vector\_numeros[i]);

end;

var

  vector\_numeros: t\_vector\_numeros;

  dimF, dimL: int16;

begin

  randomize;

  dimF:=2+random(9);

  dimL:=2+random(dimF-1);

  crear\_vector\_numeros(vector\_numeros,dimF);

  if (dimL>0) then

  begin

    cargar\_vector\_numeros(vector\_numeros,dimL);

    imprimir\_1adimL(vector\_numeros,dimL);

    imprimir\_dimLa1(vector\_numeros,dimL);

    imprimir\_dimLdiv2(vector\_numeros,dimL);

    imprimir\_general(vector\_numeros,1,dimL);

    imprimir\_general(vector\_numeros,dimL,1);

    imprimir\_general(vector\_numeros,dimL div 2,1);

    imprimir\_general(vector\_numeros,dimL div 2+1,dimL);

  end;

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.*

**(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.*

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

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

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

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

program TP4\_E4;

*{$codepage UTF8}*

uses crt;

const

  num\_total=100;

type

  t\_numero=1..num\_total;

  t\_vector\_numeros=array[t\_numero] of int16;

procedure cargar\_vector\_numeros(var vector\_numeros: t\_vector\_numeros);

var

  i: t\_numero;

begin

  for i:= 1 to num\_total do

    vector\_numeros[i]:=1+random(200);

end;

function posicion(vector\_numeros: t\_vector\_numeros; numX: int16): int16;

var

  pos: int16;

begin

  pos:=1;

  while ((pos<=num\_total) and (vector\_numeros[pos]<>numX)) do

    pos:=pos+1;

  if (pos<=num\_total) then

    posicion:=pos

  else

    posicion:=-1;

end;

procedure intercambio(var vector\_numeros: t\_vector\_numeros; numX, numY: int16);

var

  num\_aux: int16;

begin

  num\_aux:=vector\_numeros[numX];

  vector\_numeros[numX]:=vector\_numeros[numY];

  vector\_numeros[numY]:=num\_aux;

end;

function sumaVector(vector\_numeros: t\_vector\_numeros): int16;

var

  i: t\_numero;

  suma: int16;

begin

  suma:=0;

  for i:= 1 to num\_total do

    suma:=suma+vector\_numeros[i];

  sumaVector:=suma;

end;

function promedio(vector\_numeros: t\_vector\_numeros): real;

begin

  promedio:=sumaVector(vector\_numeros)/num\_total;

end;

function elementoMaximo(vector\_numeros: t\_vector\_numeros): int16;

var

  i: t\_numero;

  ele\_max, pos\_max: int16;

begin

  ele\_max:=low(int16); pos\_max:=0;

  for i:= 1 to num\_total do

    if (vector\_numeros[i]>ele\_max) then

    begin

      ele\_max:=vector\_numeros[i];

      pos\_max:=i;

    end;

  elementoMaximo:=pos\_max;

end;

function elementoMinimo(vector\_numeros: t\_vector\_numeros): int16;

var

  i: t\_numero;

  ele\_min, pos\_min: int16;

begin

  ele\_min:=high(int16); pos\_min:=0;

  for i:= 1 to num\_total do

    if (vector\_numeros[i]<ele\_min) then

    begin

      ele\_min:=vector\_numeros[i];

      pos\_min:=i;

    end;

  elementoMinimo:=pos\_min;

end;

var

  vector\_numeros: t\_vector\_numeros;

  numX, posX, posY: int16;

begin

  randomize;

  cargar\_vector\_numeros(vector\_numeros);

  numX:=1+random(200);

  textcolor(green); write('La posición del número '); textcolor(red); write(numX); textcolor(green); write(' en el vector es '); textcolor(red); writeln(posicion(vector\_numeros,numX));

  posX:=1+random(100);

  posY:=1+random(100);

  textcolor(green); write('Pre-intercambio, en las posiciones '); textcolor(red); write(posX); textcolor(green); write(' y '); textcolor(red); write(posY); textcolor(green); write(', se tienen los valores '); textcolor(red); write(vector\_numeros[posX]); textcolor(green); write(' y '); textcolor(red); write(vector\_numeros[posY]); textcolor(green); writeln(', respectivamente');

  intercambio(vector\_numeros,posX,posY);

  textcolor(green); write('Post-intercambio, en las posiciones '); textcolor(red); write(posX); textcolor(green); write(' y '); textcolor(red); write(posY); textcolor(green); write(', se tienen los valores '); textcolor(red); write(vector\_numeros[posX]); textcolor(green); write(' y '); textcolor(red); write(vector\_numeros[posY]); textcolor(green); writeln(', respectivamente');

  textcolor(green); write('La suma de todos los elementos del vector es '); textcolor(red); writeln(sumaVector(vector\_numeros));

  textcolor(green); write('El valor promedio de los elementos del vector es '); textcolor(red); writeln(promedio(vector\_numeros):0:2);

  textcolor(green); write('La posición del mayor elemento del vector es '); textcolor(red); writeln(elementoMaximo(vector\_numeros));

  textcolor(green); write('La posición del menor elemento del vector es '); textcolor(red); write(elementoMinimo(vector\_numeros));

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

  num\_total=100;

  num\_salida=0;

type

  t\_numero=1..num\_total;

  t\_vector\_numeros=array[t\_numero] of int16;

procedure cargar\_vector\_numeros(var vector\_numeros: t\_vector\_numeros; var numeros: int16);

var

  num: int16;

begin

  num:=num\_salida+random(101);

  while ((num<>num\_salida) and (numeros<num\_total)) do

  begin

    numeros:=numeros+1;

    vector\_numeros[numeros]:=num;

    num:=num\_salida+random(101);

  end;

end;

procedure elementoMaximo(vector\_numeros: t\_vector\_numeros; numeros: int16; var ele\_max, pos\_max: int16);

var

  i: t\_numero;

begin

  for i:= 1 to numeros do

    if (vector\_numeros[i]>ele\_max) then

    begin

      ele\_max:=vector\_numeros[i];

      pos\_max:=i;

    end;

end;

procedure elementoMinimo(vector\_numeros: t\_vector\_numeros; numeros: int16; var ele\_min, pos\_min: int16);

var

  i: t\_numero;

begin

  for i:= 1 to numeros do

    if (vector\_numeros[i]<ele\_min) then

    begin

      ele\_min:=vector\_numeros[i];

      pos\_min:=i;

    end;

end;

procedure intercambio(var vector\_numeros: t\_vector\_numeros; pos\_max, pos\_min: int16);

var

  num\_aux: int16;

begin

  num\_aux:=vector\_numeros[pos\_max];

  vector\_numeros[pos\_max]:=vector\_numeros[pos\_min];

  vector\_numeros[pos\_min]:=num\_aux;

end;

var

  vector\_numeros: t\_vector\_numeros;

  numeros, ele\_max, pos\_max, ele\_min, pos\_min: int16;

begin

  randomize;

  numeros:=0;

  ele\_max:=low(int16); pos\_max:=0;

  ele\_min:=high(int16); pos\_min:=0;

  cargar\_vector\_numeros(vector\_numeros,numeros);

  if (numeros>0) then

  begin

    elementoMaximo(vector\_numeros,numeros,ele\_max,pos\_max);

    elementoMinimo(vector\_numeros,numeros,ele\_min,pos\_min);

    intercambio(vector\_numeros,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;

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

  num\_total=100;

  num\_salida=0;

type

  t\_numero=1..num\_total;

  t\_vector\_numeros=array[t\_numero] of int16;

procedure cargar\_vector\_numeros(var vector\_numeros: t\_vector\_numeros; var numeros: int16);

var

  num: int16;

begin

  num:=num\_salida+random(101);

  while ((num<>num\_salida) and (numeros<num\_total)) do

  begin

    numeros:=numeros+1;

    vector\_numeros[numeros]:=num;

    num:=num\_salida+random(101);

  end;

end;

procedure elementosMaximoYMinimo(vector\_numeros: t\_vector\_numeros; numeros: int16; var ele\_max, pos\_max, ele\_min, pos\_min: int16);

var

  i: t\_numero;

begin

  for i:= 1 to numeros do

  begin

    if (vector\_numeros[i]>ele\_max) then

    begin

      ele\_max:=vector\_numeros[i];

      pos\_max:=i;

    end;

    if (vector\_numeros[i]<ele\_min) then

    begin

      ele\_min:=vector\_numeros[i];

      pos\_min:=i;

    end;

  end;

end;

procedure intercambio(var vector\_numeros: t\_vector\_numeros; pos\_max, pos\_min: int16);

var

  num\_aux: int16;

begin

  num\_aux:=vector\_numeros[pos\_max];

  vector\_numeros[pos\_max]:=vector\_numeros[pos\_min];

  vector\_numeros[pos\_min]:=num\_aux;

end;

var

  vector\_numeros: t\_vector\_numeros;

  numeros, ele\_max, pos\_max, ele\_min, pos\_min: int16;

begin

  randomize;

  numeros:=0;

  ele\_max:=low(int16); pos\_max:=0;

  ele\_min:=high(int16); pos\_min:=0;

  cargar\_vector\_numeros(vector\_numeros,numeros);

  if (numeros>0) then

  begin

    elementosMaximoYMinimo(vector\_numeros,numeros,ele\_max,pos\_max,ele\_min,pos\_min);

    intercambio(vector\_numeros,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;

end.

**Ejercicio 7.**

*Realizar un programa que lea números enteros desde teclado hasta que se ingrese el valor -1 (que no debe procesarse) e informe:*

* *la cantidad de ocurrencias de cada dígito procesado.*
* *el dígito más leído.*
* *los dígitos que no tuvieron ocurrencias.*

*Por ejemplo, si la secuencia que se lee es: 63 34 99 94 96 -1, el programa deberá informar:*

* *Número 3: 2 veces*
* *Número 4: 2 veces*
* *Número 6: 2 veces*
* *Número 9: 4 veces*
* *El dígito más leído fue el 9*
* *Los dígitos que no tuvieron ocurrencias son: 0, 1, 2, 5, 7, 8*

program TP4\_E7;

*{$codepage UTF8}*

uses crt;

const

  digitos\_total=9;

  num\_salida=-1;

  num\_corte=0;

type

  t\_digito=0..digitos\_total;

  t\_vector\_digitos=array[t\_digito] of int16;

procedure inicializar\_vector\_digitos(var vector\_digitos: t\_vector\_digitos);

var

  i: t\_digito;

begin

  for i:= 1 to digitos\_total do

    vector\_digitos[i]:=0;

end;

procedure descomponer\_numero(var vector\_digitos: t\_vector\_digitos; num: int16);

var

  digito: t\_digito;

begin

  if (num=0) then

    vector\_digitos[0]:=vector\_digitos[0]+1

  else

    while (num<>0) do

    begin

      digito:=num mod 10;

      vector\_digitos[digito]:=vector\_digitos[digito]+1;

      num:=num div 10;

    end;

end;

procedure cargar\_vector\_digitos(var vector\_digitos: t\_vector\_digitos);

var

  num: int8;

begin

  num:=num\_salida+random(high(int8));

  while (num<>num\_salida) do

  begin

    descomponer\_numero(vector\_digitos,num);

    num:=num\_salida+random(102);

  end;

end;

procedure digito\_mas\_leido(num: int16; digito: int8; var num\_max: int16; var digito\_max: int8);

begin

  if (num>num\_max) then

  begin

    num\_max:=num;

    digito\_max:=digito;

  end;

end;

procedure digitos\_sin\_ocurrencias(num: int16; digito: int8; var vector\_digitos2: t\_vector\_digitos; var dimL: int8);

begin

  if (num=num\_corte) then

  begin

    dimL:=dimL+1;

    vector\_digitos2[dimL]:=digito;

  end;

end;

procedure procesar\_vector\_digitos(vector\_digitos: t\_vector\_digitos);

var

  i: t\_digito;

  digito\_max, dimL: int8;

  num\_max: int16;

  vector\_digitos2: t\_vector\_digitos;

begin

  num\_max:=low(int16); digito\_max:=-1;

  dimL:=0;

  for i:= 0 to digitos\_total do

  begin

    textcolor(green); write('Número ',i,': '); textcolor(red); write(vector\_digitos[i]); textcolor(green); writeln(' veces');

    digito\_mas\_leido(vector\_digitos[i],i,num\_max,digito\_max);

    digitos\_sin\_ocurrencias(vector\_digitos[i],i,vector\_digitos2,dimL);

  end;

  textcolor(green); write('El dígito más leído fue el '); textcolor(red); writeln(digito\_max);

  if (dimL>0) then

  begin

  textcolor(green); write('Los dígitos que no tuvieron ocurrencias son: ');

    for i:= 1 to dimL do

    begin

      if (i<dimL) then

      begin

        textcolor(red); write(vector\_digitos2[i]); textcolor(green); write(', ');

      end

      else

      begin

        textcolor(red); write(vector\_digitos2[i]);

      end;

    end;

  end

  else

  begin

    textcolor(red); write('No hay dígitos sin ocurrencias');

  end;

end;

var

  vector\_digitos: t\_vector\_digitos;

begin

  randomize;

  inicializar\_vector\_digitos(vector\_digitos);

  cargar\_vector\_digitos(vector\_digitos);

  procesar\_vector\_digitos(vector\_digitos);

end.

**Ejercicio 8.**

*Realizar un programa que lea y almacene la información de 400 alumnos ingresantes a la Facultad de Informática de la UNLP en el año 2020. De cada alumno, se lee: número de inscripción, DNI, apellido, nombre y año de nacimiento. Una vez leída y almacenada toda la información, calcular e informar:*

* *El porcentaje de alumnos con DNI compuesto sólo por dígitos pares.*
* *Apellido y nombre de los dos alumnos de mayor edad.*

program TP4\_E8;

*{$codepage UTF8}*

uses crt;

const

  alumnos\_total=400;

type

  t\_alumno=1..alumnos\_total;

  t\_registro\_alumno=record

    numero: int16;

    dni: int32;

    apellido: string;

    nombre: string;

    nacimiento: int16;

  end;

  t\_vector\_alumnos=array[t\_alumno] of t\_registro\_alumno;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer\_alumno(var registro\_alumno: t\_registro\_alumno);

begin

  registro\_alumno.numero:=1+random(high(int16));

  registro\_alumno.dni:=10000000+random(40000001);

  registro\_alumno.apellido:=random\_string(5+random(6));

  registro\_alumno.nombre:=random\_string(5+random(6));

  registro\_alumno.nacimiento:=(2020-18)-random(10);

end;

procedure cargar\_vector\_alumnos(var vector\_alumnos: t\_vector\_alumnos);

var

  registro\_alumno: t\_registro\_alumno;

  i: t\_alumno;

begin

  for i:= 1 to alumnos\_total do

  begin

    leer\_alumno(registro\_alumno);

    vector\_alumnos[i]:=registro\_alumno;

  end;

end;

function hay\_impar(dni: int32): boolean;

var

  digito: int8;

  impar: boolean;

begin

  impar:=false;

  while ((dni<>0) and (impar<>true)) do

  begin

    digito:=dni mod 10;

    impar:=(digito mod 2<>0);

    dni:=dni div 10;

  end;

  hay\_impar:=impar;

end;

procedure actualizar\_minimos(nacimiento: int16; apellido, nombre: string ; var nacimiento\_min1, nacimiento\_min2: int16; var apellido\_min1, nombre\_min1, apellido\_min2, nombre\_min2: string);

begin

  if (nacimiento<nacimiento\_min1) then

  begin

    nacimiento\_min2:=nacimiento\_min1;

    apellido\_min2:=apellido\_min1;

    nombre\_min2:=nombre\_min1;

    nacimiento\_min1:=nacimiento;

    apellido\_min1:=apellido;

    nombre\_min1:=nombre;

  end

  else

    if (nacimiento<nacimiento\_min2) then

    begin

      nacimiento\_min2:=nacimiento;

      apellido\_min2:=apellido;

      nombre\_min2:=nombre;

    end;

end;

procedure procesar\_vector\_alumnos(vector\_alumnos: t\_vector\_alumnos; var porcentaje\_pares: real; var apellido\_min1, nombre\_min1, apellido\_min2, nombre\_min2: string);

var

  i: t\_alumno;

  alumnos\_pares, nacimiento\_min1, nacimiento\_min2: int16;

begin

  alumnos\_pares:=0;

  nacimiento\_min1:=high(int16); nacimiento\_min2:=high(int16);

  for i:= 1 to alumnos\_total do

  begin

    if (hay\_impar(vector\_alumnos[i].dni)=false) then

      alumnos\_pares:=alumnos\_pares+1;

    actualizar\_minimos(vector\_alumnos[i].nacimiento,vector\_alumnos[i].apellido,vector\_alumnos[i].nombre,nacimiento\_min1,nacimiento\_min2,apellido\_min1,nombre\_min1,apellido\_min2,nombre\_min2);

  end;

  porcentaje\_pares:=alumnos\_pares/alumnos\_total\*100;

end;

var

  vector\_alumnos: t\_vector\_alumnos;

  porcentaje\_pares: real;

  apellido\_min1, nombre\_min1, apellido\_min2, nombre\_min2: string;

begin

  randomize;

  porcentaje\_pares:=0;

  apellido\_min1:=''; nombre\_min1:=''; apellido\_min2:=''; nombre\_min2:='';

  cargar\_vector\_alumnos(vector\_alumnos);

  procesar\_vector\_alumnos(vector\_alumnos,porcentaje\_pares,apellido\_min1,nombre\_min1,apellido\_min2,nombre\_min2);

  textcolor(green); write('El porcentaje de alumnos con DNI compuesto sólo por dígitos pares es '); textcolor(red); write(porcentaje\_pares:0:2); textcolor(green); writeln('%');

  textcolor(green); write('El apellido y nombre de los dos alumnos de mayor edad son '); textcolor(red); write(apellido\_min1,' ',nombre\_min1); textcolor(green); write(' y '); textcolor(red); write(apellido\_min2,' ',nombre\_min2);

end.

**Ejercicio 9.**

*Modificar la solución del punto anterior considerando que el programa lea y almacene la información de, a lo sumo, 400 alumnos. La lectura finaliza cuando se ingresa el DNI -1 (que no debe procesarse).*

program TP4\_E9;

*{$codepage UTF8}*

uses crt;

const

  alumnos\_total=400;

  dni\_salida=-1;

type

  t\_alumno=1..alumnos\_total;

  t\_registro\_alumno=record

    numero: int16;

    dni: int32;

    apellido: string;

    nombre: string;

    nacimiento: int16;

  end;

  t\_vector\_alumnos=array[t\_alumno] of t\_registro\_alumno;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer\_alumno(var registro\_alumno: t\_registro\_alumno);

var

  i: int8;

begin

  i:=random(100);

  if (i=0) then

    registro\_alumno.dni:=dni\_salida

  else

    registro\_alumno.dni:=10000000+random(40000001);

  if (registro\_alumno.dni<>dni\_salida) then

  begin

    registro\_alumno.numero:=1+random(high(int16));

    registro\_alumno.apellido:=random\_string(5+random(6));

    registro\_alumno.nombre:=random\_string(5+random(6));

    registro\_alumno.nacimiento:=(2020-18)-random(10);

  end;

end;

procedure cargar\_vector\_alumnos(var vector\_alumnos: t\_vector\_alumnos; var alumnos: int16);

var

  registro\_alumno: t\_registro\_alumno;

begin

  leer\_alumno(registro\_alumno);

  while ((registro\_alumno.dni<>dni\_salida) and (alumnos<alumnos\_total)) do

  begin

    alumnos:=alumnos+1;

    vector\_alumnos[alumnos]:=registro\_alumno;

    leer\_alumno(registro\_alumno);

  end;

end;

function hay\_impar(dni: int32): boolean;

var

  digito: int8;

  impar: boolean;

begin

  impar:=false;

  while ((dni<>0) and (impar<>true)) do

  begin

    digito:=dni mod 10;

    impar:=(digito mod 2<>0);

    dni:=dni div 10;

  end;

  hay\_impar:=impar;

end;

procedure actualizar\_minimos(nacimiento: int16; apellido, nombre: string ; var nacimiento\_min1, nacimiento\_min2: int16; var apellido\_min1, nombre\_min1, apellido\_min2, nombre\_min2: string);

begin

  if (nacimiento<nacimiento\_min1) then

  begin

    nacimiento\_min2:=nacimiento\_min1;

    apellido\_min2:=apellido\_min1;

    nombre\_min2:=nombre\_min1;

    nacimiento\_min1:=nacimiento;

    apellido\_min1:=apellido;

    nombre\_min1:=nombre;

  end

  else

    if (nacimiento<nacimiento\_min2) then

    begin

      nacimiento\_min2:=nacimiento;

      apellido\_min2:=apellido;

      nombre\_min2:=nombre;

    end;

end;

procedure procesar\_vector\_alumnos(vector\_alumnos: t\_vector\_alumnos; alumnos: int16; var porcentaje\_pares: real; var apellido\_min1, nombre\_min1, apellido\_min2, nombre\_min2: string);

var

  i: t\_alumno;

  alumnos\_pares, nacimiento\_min1, nacimiento\_min2: int16;

begin

  alumnos\_pares:=0;

  nacimiento\_min1:=high(int16); nacimiento\_min2:=high(int16);

  for i:= 1 to alumnos do

  begin

    if (hay\_impar(vector\_alumnos[i].dni)=false) then

      alumnos\_pares:=alumnos\_pares+1;

    actualizar\_minimos(vector\_alumnos[i].nacimiento,vector\_alumnos[i].apellido,vector\_alumnos[i].nombre,nacimiento\_min1,nacimiento\_min2,apellido\_min1,nombre\_min1,apellido\_min2,nombre\_min2);

  end;

  porcentaje\_pares:=alumnos\_pares/alumnos\*100;

end;

var

  vector\_alumnos: t\_vector\_alumnos;

  alumnos: int16;

  porcentaje\_pares: real;

  apellido\_min1, nombre\_min1, apellido\_min2, nombre\_min2: string;

begin

  randomize;

  alumnos:=0;

  porcentaje\_pares:=0;

  apellido\_min1:=''; nombre\_min1:=''; apellido\_min2:=''; nombre\_min2:='';

  cargar\_vector\_alumnos(vector\_alumnos,alumnos);

  if (alumnos>0) then

  begin

    procesar\_vector\_alumnos(vector\_alumnos,alumnos,porcentaje\_pares,apellido\_min1,nombre\_min1,apellido\_min2,nombre\_min2);

    textcolor(green); write('El porcentaje de alumnos con DNI compuesto sólo por dígitos pares es '); textcolor(red); write(porcentaje\_pares:0:2); textcolor(green); writeln('%');

    textcolor(green); write('El apellido y nombre de los dos alumnos de mayor edad son '); textcolor(red); write(apellido\_min1,' ',nombre\_min1); textcolor(green); write(' y '); textcolor(red); write(apellido\_min2,' ',nombre\_min2);

  end;

end.

**Ejercicio 10.**

*Realizar un programa que lea y almacene el salario de los empleados de una empresa de turismo (a lo sumo, 300 empleados). La carga finaliza cuando se lea el salario 0 (que no debe procesarse) o cuando se completa el vector. Una vez finalizada la carga de datos, se pide:*

**(a)** *Realizar un módulo que incremente el salario de cada empleado en un 15%. Para ello, implementar un módulo que reciba como parámetro un valor real X, el vector de valores reales y su dimensión lógica y retorne el mismo vector en el cual cada elemento fue multiplicado por el valor X.*

**(b)** *Realizar un módulo que muestre en pantalla el sueldo promedio de los empleados de la empresa.*

program TP4\_E10;

*{$codepage UTF8}*

uses crt;

const

  empleados\_total=300;

  salario\_salida=0;

  incremento=1.15;

type

  t\_empleado=1..empleados\_total;

  t\_vector\_salarios=array[t\_empleado] of real;

procedure inicializar\_vector\_salarios(var vector\_salarios: t\_vector\_salarios);

var

  i: t\_empleado;

begin

  for i:= 1 to empleados\_total do

    vector\_salarios[i]:=0;

end;

procedure cargar\_vector\_salarios(var vector\_salarios: t\_vector\_salarios; var empleados: int16);

var

  salario: real;

begin

  salario:=salario\_salida+random(1001)/10;

  while ((salario<>salario\_salida) and (empleados<empleados\_total)) do

  begin

    empleados:=empleados+1;

    vector\_salarios[empleados]:=salario;

    salario:=salario\_salida+random(1001)/10;

  end;

end;

procedure incrementar\_salarios(incremento: real; var vector\_salarios: t\_vector\_salarios; empleados: int16);

var

  i: t\_empleado;

begin

  for i:= 1 to empleados do

    vector\_salarios[i]:=vector\_salarios[i]\*incremento;

end;

procedure calcular\_salario\_promedio(vector\_salarios: t\_vector\_salarios; empleados: int16);

var

  i: t\_empleado;

  salario\_total, salario\_prom: real;

begin

  salario\_total:=0;

  for i:= 1 to empleados do

    salario\_total:=salario\_total+vector\_salarios[i];

  salario\_prom:=salario\_total/empleados;

  textcolor(green); write('El sueldo promedio de los empleados de la empresa es $'); textcolor(red); write(salario\_prom:0:2);

end;

var

  vector\_salarios: t\_vector\_salarios;

  empleados: int16;

begin

  randomize;

  empleados:=0;

  inicializar\_vector\_salarios(vector\_salarios);

  cargar\_vector\_salarios(vector\_salarios,empleados);

  if (empleados>0) then

  begin

    incrementar\_salarios(incremento,vector\_salarios,empleados);

    calcular\_salario\_promedio(vector\_salarios,empleados);

  end;

end.

**Ejercicio 11.**

*El colectivo de fotógrafos ArgenPics desea conocer los gustos de sus seguidores en las redes sociales. Para ello, para cada una de las 200 fotos publicadas en su página de Facebook, cuenta con la siguiente información: título de la foto, el autor de la foto, cantidad de Me gusta, cantidad de clics y cantidad de comentarios de usuarios. Realizar un programa que lea y almacene esta información. Una vez finalizada la lectura, el programa debe procesar los datos e informar:*

* *Título de la foto más vista (la que posee mayor cantidad de clics).*
* *Cantidad total de Me gusta recibidos a las fotos cuyo autor es el fotógrafo “Art Vandelay”.*
* *Cantidad de comentarios recibidos para cada una de las fotos publicadas en esa página.*

program TP4\_E11;

*{$codepage UTF8}*

uses crt;

const

  fotos\_total=200;

  autor\_corte='Art Vandelay';

type

  t\_foto=1..fotos\_total;

  t\_registro\_foto=record

    titulo: string;

    autor: string;

    megusta: int16;

    clics: int16;

    comentarios: int16;

  end;

  t\_vector\_fotos=array[t\_foto] of t\_registro\_foto;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer\_foto(var registro\_foto: t\_registro\_foto);

var

  i: int8;

begin

  registro\_foto.titulo:=random\_string(5+random(6));

  i:=random(10);

  if (i=0) then

    registro\_foto.autor:=autor\_corte

  else

    registro\_foto.autor:=random\_string(5+random(6));

  registro\_foto.megusta:=random(10001);

  registro\_foto.clics:=random(10001);

  registro\_foto.comentarios:=random(10001);

end;

procedure cargar\_vector\_fotos(var vector\_fotos: t\_vector\_fotos);

var

  registro\_foto: t\_registro\_foto;

  i: t\_foto;

begin

  for i:= 1 to fotos\_total do

  begin

    leer\_foto(registro\_foto);

    vector\_fotos[i]:=registro\_foto;

  end;

end;

procedure actualizar\_maximo(clics: int16; titulo: string; var clics\_max: int16; var titulo\_max: string);

begin

  if (clics>clics\_max) then

  begin

    clics\_max:=clics;

    titulo\_max:=titulo;

  end;

end;

procedure procesar\_vector\_fotos(vector\_fotos: t\_vector\_fotos; var titulo\_max: string; var megusta\_corte: int16);

var

  i: t\_foto;

  clics\_max: int16;

begin

  clics\_max:=low(int16);

  for i:= 1 to fotos\_total do

  begin

    actualizar\_maximo(vector\_fotos[i].clics,vector\_fotos[i].titulo,clics\_max,titulo\_max);

    if (vector\_fotos[i].autor=autor\_corte) then

      megusta\_corte:=megusta\_corte+1;

    textcolor(green); write('La cantidad de comentarios recibidos de la foto '); textcolor(red); write(vector\_fotos[i].titulo); textcolor(green); write(' es '); textcolor(red); writeln(vector\_fotos[i].comentarios);

  end;

end;

var

  vector\_fotos: t\_vector\_fotos;

  megusta\_corte: int16;

  titulo\_max: string;

begin

  randomize;

  titulo\_max:='';

  megusta\_corte:=0;

  cargar\_vector\_fotos(vector\_fotos);

  procesar\_vector\_fotos(vector\_fotos,titulo\_max,megusta\_corte);

  textcolor(green); write('El título de la foto más vista (la que posee mayor cantidad de clics) es '); textcolor(red); writeln(titulo\_max);

  textcolor(green); write('La cantidad total de Me gusta recibidas a las fotos cuyo autor es el fotógrafo "'); textcolor(yellow); write(autor\_corte); textcolor(green); write('" es '); textcolor(red); write(megusta\_corte);

end.

**Ejercicio 12.**

*En astrofísica, una galaxia se identifica por su nombre, su tipo (1. elíptica; 2. espiral; 3. lenticular; 4. irregular), su masa (medida en kg) y la distancia en pársecs (pc) medida desde la Tierra. La Unión Astronómica Internacional cuenta con datos correspondientes a las 53 galaxias que componen el Grupo Local. Realizar un programa que lea y almacene estos datos y, una vez finalizada la carga, informe:*

* *La cantidad de galaxias de cada tipo.*
* *La masa total acumulada de las 3 galaxias principales (la Vía Láctea, Andrómeda y Triángulo) y el porcentaje que esto representa respecto a la masa de todas las galaxias.*
* *La cantidad de galaxias no irregulares que se encuentran a menos de 1000 pc.*
* *El nombre de las dos galaxias con mayor masa y el de las dos galaxias con menor masa.*

program TP4\_E12;

*{$codepage UTF8}*

uses crt;

const

  galaxias\_total=53;

  tipo\_ini=1; tipo\_fin=4;

  galaxia\_corte1='la Via Lactea'; galaxia\_corte2='Andromeda'; galaxia\_corte3='Triangulo';

  tipo\_corte=4; distancia\_corte=1000.0;

type

  t\_galaxia=1..galaxias\_total;

  t\_tipo=tipo\_ini..tipo\_fin;

  t\_registro\_galaxia=record

    nombre: string;

    tipo: t\_tipo;

    masa: real;

    distancia: real;

  end;

  t\_vector\_galaxias=array[t\_galaxia] of t\_registro\_galaxia;

  t\_vector\_tipos=array[t\_tipo] of int16;

procedure inicializar\_vector\_tipos(var vector\_tipos: t\_vector\_tipos);

var

  i: t\_tipo;

begin

  for i:= tipo\_ini to tipo\_fin do

    vector\_tipos[i]:=0;

end;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer\_galaxia(var registro\_galaxia: t\_registro\_galaxia);

var

  i: int8;

begin

  i:=random(4);

  if (i=0) then

    registro\_galaxia.nombre:=galaxia\_corte1

  else if (i=1) then

    registro\_galaxia.nombre:=galaxia\_corte2

  else if (i=2) then

    registro\_galaxia.nombre:=galaxia\_corte3

  else

    registro\_galaxia.nombre:='Galaxia '+random\_string(5+random(6));

  registro\_galaxia.tipo:=tipo\_ini+random(tipo\_fin);

  registro\_galaxia.masa:=11+random(99991)/10;

  registro\_galaxia.masa:=11+random(99991)/10;

end;

procedure cargar\_vector\_galaxias(var vector\_galaxias: t\_vector\_galaxias);

var

  registro\_galaxia: t\_registro\_galaxia;

  i: t\_galaxia;

begin

  for i:= 1 to galaxias\_total do

  begin

    leer\_galaxia(registro\_galaxia);

    vector\_galaxias[i]:=registro\_galaxia;

  end;

end;

procedure actualizar\_maximos(masa: real; nombre: string; var masa\_max1, masa\_max2: real; var nombre\_max1, nombre\_max2: string);

begin

  if (masa>masa\_max1) then

  begin

    masa\_max2:=masa\_max1;

    nombre\_max2:=nombre\_max1;

    masa\_max1:=masa;

    nombre\_max1:=nombre;

  end

  else

    if (masa>masa\_max2) then

    begin

      masa\_max2:=masa;

      nombre\_max2:=nombre;

    end;

end;

procedure actualizar\_minimos(masa: real; nombre: string; var masa\_min1, masa\_min2: real; var nombre\_min1, nombre\_min2: string);

begin

  if (masa<masa\_min1) then

  begin

    masa\_min2:=masa\_min1;

    nombre\_min2:=nombre\_min1;

    masa\_min1:=masa;

    nombre\_min1:=nombre;

  end

  else

    if (masa<masa\_min2) then

    begin

      masa\_min2:=masa;

      nombre\_min2:=nombre;

    end;

end;

procedure procesar\_vector\_galaxias(vector\_galaxias: t\_vector\_galaxias; var vector\_tipos: t\_vector\_tipos; var masa\_principales, porcentaje\_principales: real; var galaxias\_corte: int8; var nombre\_max1, nombre\_max2, nombre\_min1, nombre\_min2: string);

var

  i: t\_galaxia;

  masa\_total, masa\_max1, masa\_max2, masa\_min1, masa\_min2: real;

begin

  masa\_total:=0;

  masa\_max1:=-9999999; masa\_max2:=-9999999;

  masa\_min1:=9999999; masa\_min2:=9999999;

  for i:= 1 to galaxias\_total do

  begin

    vector\_tipos[vector\_galaxias[i].tipo]:=vector\_tipos[vector\_galaxias[i].tipo]+1;

    masa\_total:=masa\_total+vector\_galaxias[i].masa;

    if ((vector\_galaxias[i].nombre=galaxia\_corte1) or (vector\_galaxias[i].nombre=galaxia\_corte2) or (vector\_galaxias[i].nombre=galaxia\_corte3)) then

      masa\_principales:=masa\_principales+vector\_galaxias[i].masa;

    if ((vector\_galaxias[i].tipo<>tipo\_corte) and (vector\_galaxias[i].distancia<distancia\_corte)) then

      galaxias\_corte:=galaxias\_corte+1;

    actualizar\_maximos(vector\_galaxias[i].masa,vector\_galaxias[i].nombre,masa\_max1,masa\_max2,nombre\_max1,nombre\_max2);

    actualizar\_minimos(vector\_galaxias[i].masa,vector\_galaxias[i].nombre,masa\_min1,masa\_min2,nombre\_min1,nombre\_min2);

  end;

  porcentaje\_principales:=masa\_principales/masa\_total\*100;

end;

var

  vector\_tipos\_string: array[t\_tipo] of string=('eliptica', 'espiral', 'lenticular', 'irregular');

  vector\_galaxias: t\_vector\_galaxias;

  vector\_tipos: t\_vector\_tipos;

  galaxias\_corte: int8;

  masa\_principales, porcentaje\_principales: real;

  nombre\_max1, nombre\_max2, nombre\_min1, nombre\_min2: string;

begin

  randomize;

  inicializar\_vector\_tipos(vector\_tipos);

  masa\_principales:=0; porcentaje\_principales:=0;

  galaxias\_corte:=0;

  nombre\_max1:=''; nombre\_max2:='';

  nombre\_min1:=''; nombre\_min2:='';

  cargar\_vector\_galaxias(vector\_galaxias);

  procesar\_vector\_galaxias(vector\_galaxias,vector\_tipos,masa\_principales,porcentaje\_principales,galaxias\_corte,nombre\_max1,nombre\_max2,nombre\_min1,nombre\_min2);

  textcolor(green); write('La cantidad de galaxias de cada tipo (1. elíptica; 2. espiral; 3. lenticular; 4. irregular) es '); textcolor(red); write(vector\_tipos[1],', ',vector\_tipos[2],', ',vector\_tipos[3],', ',vector\_tipos[4]); textcolor(green); writeln(', respectivamente');

  textcolor(green); write('La masa total acumulada de las 3 galaxias principales ('); textcolor(yellow); write(galaxia\_corte1,', ',galaxia\_corte2,', ',galaxia\_corte3); textcolor(green); write(') y el porcentaje que esto representa respecto a la masa de todas las galaxias son '); textcolor(red); write(masa\_principales:0:2); textcolor(green); write(' y '); textcolor(red); write(porcentaje\_principales:0:2); textcolor(green); writeln('%, respectivamente');

  textcolor(green); write('La cantidad de galaxias no '); textcolor(yellow); write(vector\_tipos\_string[tipo\_corte]); textcolor(green); write(' que se encuentran a menos de '); textcolor(yellow); write(distancia\_corte:0:2); textcolor(green); write(' pc es '); textcolor(red); writeln(galaxias\_corte);

  textcolor(green); write('Los nombres de las dos galaxias con mayor masa son '); textcolor(red); write(nombre\_max1); textcolor(green); write(' y '); textcolor(red); writeln(nombre\_max2);

  textcolor(green); write('Los nombres de las dos galaxias con menor masa son '); textcolor(red); write(nombre\_min1); textcolor(green); write(' y '); textcolor(red); write(nombre\_min2);

end.

**Ejercicio 13.**

*El Grupo Intergubernamental de Expertos sobre el Cambio Climático de la ONU (IPCC) realiza todos los años mediciones de temperatura en 100 puntos diferentes del planeta y, para cada uno de estos lugares, obtiene un promedio anual. Este relevamiento se viene realizando desde hace 50 años y, con todos los datos recolectados, el IPCC se encuentra en condiciones de realizar análisis estadísticos. Realizar un programa que lea y almacene los datos correspondientes a las mediciones de los últimos 50 años (la información se ingresa ordenada por año). Una vez finalizada la carga de la información, obtener:*

* *El año con mayor temperatura promedio a nivel mundial.*
* *El año con la mayor temperatura detectada en algún punto del planeta en los últimos 50 años.*

program TP4\_E13;

*{$codepage UTF8}*

uses crt;

const

  puntos\_total=100;

  anio\_ini=1974; anio\_fin=2023;

type

  t\_punto=1..puntos\_total;

  t\_anio=anio\_ini..anio\_fin;

  t\_vector\_puntos=array[t\_punto] of real;

  t\_vector\_anios=array[t\_anio] of t\_vector\_puntos;

procedure inicializar\_vector\_anios(var vector\_anios: t\_vector\_anios);

var

  i: t\_anio;

  j: t\_punto;

begin

  for i:= anio\_ini to anio\_fin do

    for j:= 1 to puntos\_total do

      vector\_anios[i][j]:=0;

end;

procedure cargar\_vector\_anios(var vector\_anios: t\_vector\_anios);

var

  i: t\_anio;

  j: t\_punto;

begin

  for i:= anio\_ini to anio\_fin do

    for j:= 1 to puntos\_total do

      vector\_anios[i][j]:=-50+random(1001)/10;

end;

procedure actualizar\_maximo1(promedio\_anio: real; anio: int16; var promedio\_max: real; var anio\_max1: int16);

begin

  if (promedio\_anio>promedio\_max) then

  begin

    promedio\_max:=promedio\_anio;

    anio\_max1:=anio;

  end;

end;

procedure actualizar\_maximo2(temperatura: real; anio: int16; var temperatura\_max: real; var anio\_max2: int16);

begin

  if (temperatura>temperatura\_max) then

  begin

    temperatura\_max:=temperatura;

    anio\_max2:=anio;

  end;

end;

procedure procesar\_vector\_anios(vector\_anios: t\_vector\_anios; var anio\_max1, anio\_max2: int16);

var

  i: t\_anio;

  j: t\_punto;

  temperatura\_anio, promedio\_anio, promedio\_max, temperatura\_max: real;

begin

  promedio\_max:=-9999999;

  temperatura\_max:=-9999999;

  for i:= anio\_ini to anio\_fin do

  begin

    temperatura\_anio:=0;

    for j:= 1 to puntos\_total do

    begin

      temperatura\_anio:=temperatura\_anio+vector\_anios[i][j];

      actualizar\_maximo2(vector\_anios[i][j],i,temperatura\_max,anio\_max2);

    end;

    promedio\_anio:=temperatura\_anio/puntos\_total;

    actualizar\_maximo1(promedio\_anio,i,promedio\_max,anio\_max1);

  end;

end;

var

  vector\_anios: t\_vector\_anios;

  anio\_max1, anio\_max2: int16;

begin

  randomize;

  anio\_max1:=0; anio\_max2:=0;

  inicializar\_vector\_anios(vector\_anios);

  cargar\_vector\_anios(vector\_anios);

  procesar\_vector\_anios(vector\_anios,anio\_max1,anio\_max2);

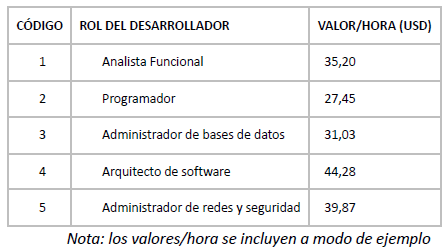
  textcolor(green); write('El año con mayor temperatura promedio a nivel mundial es '); textcolor(red); writeln(anio\_max1);

  textcolor(green); write('El año con la mayor temperatura detectada en algún punto del planeta en los últimos 50 años es '); textcolor(red); write(anio\_max2);

end.

**Ejercicio 14.**

*El repositorio de código fuente más grande en la actualidad, GitHub, desea estimar el monto invertido en los proyectos que aloja. Para ello, dispone de una tabla con información de los desarrolladores que participan en un proyecto de software, junto al valor promedio que se paga por hora de trabajo:*



*Realizar un programa que procese la información de los desarrolladores que participaron en los 1000 proyectos de software más activos durante el año 2017. De cada participante, se conoce su país de residencia, código de proyecto (1 a 1000), el nombre del proyecto en el que participó, el rol que cumplió en dicho proyecto (1 a 5) y la cantidad de horas trabajadas. La lectura finaliza al ingresar el código de proyecto -1, que no debe procesarse. Al finalizar la lectura, el programa debe informar:*

* *El monto total invertido en desarrolladores con residencia en Argentina.*
* *La cantidad total de horas trabajadas por Administradores de bases de datos.*
* *El código del proyecto con menor monto invertido.*
* *La cantidad de Arquitectos de software de cada proyecto.*

program TP4\_E14;

*{$codepage UTF8}*

uses crt;

const

  proyecto\_ini=1; proyecto\_fin=1000;

  rol\_ini=1; rol\_fin=5;

  proyecto\_salida=-1;

  pais\_corte='Argentina';

  rol\_corte1=3;

  rol\_corte2=4;

type

  t\_proyecto=proyecto\_ini..proyecto\_fin;

  t\_rol=rol\_ini..rol\_fin;

   t\_registro\_participante=record

    pais: string;

    proyecto: int16;

    nombre: string;

    rol: t\_rol;

    horas: int16;

  end;

  t\_registro\_proyecto=record

    monto: real;

    cantidad: int16;

  end;

  t\_vector\_proyectos=array[t\_proyecto] of t\_registro\_proyecto;

  t\_vector\_salarios=array[t\_rol] of real;

procedure cargar\_vector\_salarios(var vector\_salarios: t\_vector\_salarios);

begin

  vector\_salarios[1]:=35.20;

  vector\_salarios[2]:=27.45;

  vector\_salarios[3]:=31.03;

  vector\_salarios[4]:=44.28;

  vector\_salarios[5]:=39.87;

end;

procedure inicializar\_vector\_proyectos(var vector\_proyectos: t\_vector\_proyectos);

var

  i: t\_proyecto;

begin

  for i:= proyecto\_ini to proyecto\_fin do

  begin

    vector\_proyectos[i].monto:=0;

    vector\_proyectos[i].cantidad:=0;

  end;

end;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer\_participante(var registro\_participante: t\_registro\_participante);

var

  i: int8;

begin

  i:=random(100);

  if (i=0) then

    registro\_participante.proyecto:=proyecto\_salida

  else

    registro\_participante.proyecto:=proyecto\_ini+random(proyecto\_fin);

  if (registro\_participante.proyecto<>proyecto\_salida) then

  begin

    i:=random(10);

    if (i=0) then

      registro\_participante.pais:=pais\_corte

    else

      registro\_participante.pais:=random\_string(5+random(6));

    registro\_participante.nombre:=random\_string(5+random(6));

    registro\_participante.rol:=rol\_ini+random(rol\_fin);

    registro\_participante.horas:=1+random(100);

  end;

end;

procedure cargar\_vector\_proyectos(var vector\_proyectos: t\_vector\_proyectos; var monto\_corte: real; var horas\_corte: int16; vector\_salarios: t\_vector\_salarios);

var

  registro\_participante: t\_registro\_participante;

begin

  leer\_participante(registro\_participante);

  while (registro\_participante.proyecto<>proyecto\_salida) do

  begin

    if (registro\_participante.pais=pais\_corte) then

      monto\_corte:=monto\_corte+vector\_salarios[registro\_participante.rol]\*registro\_participante.horas;

    if (registro\_participante.rol=rol\_corte1) then

      horas\_corte:=horas\_corte+registro\_participante.horas;

    vector\_proyectos[registro\_participante.proyecto].monto:=vector\_proyectos[registro\_participante.proyecto].monto+vector\_salarios[registro\_participante.rol]\*registro\_participante.horas;

    if (registro\_participante.rol=rol\_corte2) then

      vector\_proyectos[registro\_participante.proyecto].cantidad:=vector\_proyectos[registro\_participante.proyecto].cantidad+1;

    leer\_participante(registro\_participante);

  end;

end;

procedure procesar\_vector\_proyectos(vector\_proyectos: t\_vector\_proyectos; var proyecto\_min: int16);

var

  i: t\_proyecto;

  monto\_min: real;

begin

  monto\_min:=9999999;

  for i:= proyecto\_ini to proyecto\_fin do

  begin

    if ((vector\_proyectos[i].monto>0) and (vector\_proyectos[i].monto<monto\_min)) then

    begin

      monto\_min:=vector\_proyectos[i].monto;

      proyecto\_min:=i;

    end;

    if (vector\_proyectos[i].cantidad>0) then

    begin

      textcolor(green); write('La cantidad de Arquitectos de software del proyecto ',i,' es '); textcolor(red); writeln(vector\_proyectos[i].cantidad);

    end;

  end;

end;

var

  vector\_salarios: t\_vector\_salarios;

  vector\_proyectos: t\_vector\_proyectos;

  horas\_corte, proyecto\_min: int16;

  monto\_corte: real;

begin

  randomize;

  cargar\_vector\_salarios(vector\_salarios);

  monto\_corte:=0;

  horas\_corte:=0;

  proyecto\_min:=0;

  inicializar\_vector\_proyectos(vector\_proyectos);

  cargar\_vector\_proyectos(vector\_proyectos,monto\_corte,horas\_corte,vector\_salarios);

  textcolor(green); write('El monto total invertido en desarrolladores con residencia en '); textcolor(yellow); write(pais\_corte); textcolor(green); write(' es U$D '); textcolor(red); writeln(monto\_corte:0:2);

  textcolor(green); write('La cantidad total de horas trabajadas por Administradores de bases de datos es '); textcolor(red); writeln(horas\_corte);

  procesar\_vector\_proyectos(vector\_proyectos,proyecto\_min);

  textcolor(green); write('El código de proyecto con menor monto invertido es '); textcolor(red); write(proyecto\_min);

end.