Práctica 5

Estructura de Datos Vector (parte 2)

Algoritmos y Programación 1
Ciencia de Datos en Organizaciones

2025

Temas de la Práctica 5

- Contenidos
  - **Operaciones en Vectores**
    - Búsqueda
    - Orden
    - Procesamiento

Ejercicio 2 a

Dado un vector de enteros de 500 valores enteros, realice un módulo que reciba dicho vector y un valor n y retorne si n se encuentra en el vector o no.

```
Program uno;
const
   TAM = 500;
type
   vector = array [1..TAM]  of integer;
var
  v: vector;
  num:integer;
begin
   read (num);
   cargarNumeros (v);
   if (buscar(v,num) = true)  then
            writeln (num, ‘ está en el vector’)
   else
            writeln (num, ‘ no está en el vector’);
end.
```

```
Function buscar (v:vector; n:integer): boolean;
var
  pos:integer;
  esta:boolean;
begin
  esta:= false;
  pos:=1;
  while ( (pos <= TAM ) and (not esta) ) do begin
    if (v[pos]= n) then
            esta:=true
    else
            pos:= pos + 1;
   end;
   buscar := esta;
end;
```

Ejercicio 2 b

Modifique el módulo del inciso a. considerando ahora que el vector se encuentra ordenado de manera ascendente.

```
Program uno;
const
   TAM = 500;
type
   vector = array [1..TAM]  of integer;
var
  v: vector;
  num:integer;
begin
   read (num);
   cargarNumeros (v);
   if (buscarOrdenado (v,num)) then
       writeln (num, ‘ está en el vector’)
   else
       writeln (num, ‘ no está en el vector’);
end.
```

```
Function buscarOrdenado (v:vector; n:integer): boolean;
var
  pos:integer;
  esta:boolean;
begin
  pos:=1;
  while ( (pos <= TAM) and (v[pos]<n)) do
        pos:= pos + 1;
  if ( (pos <= TAM) and (v[pos]= n))then
             esta := true
   else
             esta:= false;
  buscarOrdenado:=esta;
end;
```

Ejercicio 5

Una empresa de transporte de caudales desea optimizar el servicio que brinda a sus clientes. Para ello, lee información sobre todos los viajes realizados durante el mes de marzo. De cada viaje lee la siguiente información: día del mes (de 1 a 31), monto de dinero transportado y distancia recorrida por el camión (medida en kilómetros). Realizar un programa que lea y almacene la información de los 160 viajes realizados. Realizar un módulo que reciba el vector generado e informe:

a. El monto promedio de dinero transportado de los viajes realizados
b. La distancia recorrida y el día del mes en que se realizó el viaje que transportó menos dinero.
c. La cantidad de viajes realizados cada día del mes.

```
Program cinco;
const
   TAM = 160;
Type
  viaje =  record
     dia: integer;
     monto: real;
     distancia: real;
   end;
   vector = array [1..TAM]  of viaje;
var
  v: vector;
Begin
   cargarViajes(v);
   ...
end.
```

```
Procedure cargarViajes(var v:vector);
var
 i:integer;
 unviaje: viaje;
begin
    For i:= 1 to TAM do begin
       LeerViaje(unviaje);
       v[i]:= unviaje;
    end;
end;
```

```
Procedure LeerViaje(var v:viaje);
begin
   writeln('Ingrese distancia del viaje en KM ');
   readln(v.distancia);
   writeln('Ingrese día del viaje');
   readln(v.dia);
   writeln('Ingrese monto');
   readln(v.monto);
end;
```

Es lo mismo que solo escribir LeerViaje(v[i])

Ejercicio 5

El monto promedio de dinero transportado de los viajes realizados
La distancia recorrida y el día del mes en que se realizó el viaje que transportó menos dinero.
La cantidad de viajes realizados cada día del mes.

```
Program cinco;
const
   TAM = 160;
Type
   viajes =  record
     dia:integer;
     monto:real;
     distancia: real;
   end;
   vector = array [1..TAM]  of viajes;
   vectordias = array [1..31]  of integer;
var
  v: vector;
Begin
   cargarViajes(v);
   InformeViajes(v);
end.
```

```
Procedure InformeViajes (v:vector);
var
   i, diaMin:integer;
   vdias: vectordias;
   suma,montomin ,distancia:real;
Begin
    suma:=0;
    InicializarVector(vdias);
    diaMin:=99 ; montoMin:=9999;
    For i := 1 to TAM do begin
        suma:=suma+v[i].monto;
        If (v[i].monto<montoMin) then begin
            montoMin:=v[i].monto;
            diaMin:= v[i].dia;
            distancia:= v[i].distancia;
        end;
        vdias[v[i].dia]:=vdias[v[i].dia]+1;
   end;
   writeln ('El promedio de dinero transportado fue: ' ,suma/TAM );
   writeln ('El día que se transportó menos dinero fue : ' , diaMin, 'la distancia recorrida fue ', distancia);
   Viajespordia(vdias);
end;
```

```
procedure InicializarVector (var v:vectordias);
var i :integer;
begin
    For i :=1 to 31 do
        v[i]:= 0;
end;
```

```
procedure Viajespordia(v:vectordias);
var i :integer;
begin
   for i :=1 to 31 do
    writeln ('La cantidad de viajes realizadas el día ',i, ' fue ', v[i]);
end;
```

Modularizar el cálculo de mínimo

Ejercicio 5

```
procedure calcularMínimo (vi:viaje; var min: vector);
begin
  If (vi.monto<min.monto) then begin
        min:= vi;
    end;
end;
```

```
Procedure InformeViajes (v:vector);
var
  i:integer;    min: vector;
  vdias: vectordias;
  suma:real;
Begin
   suma:=0;
   InicializarVector(vdias);
   min.dia:=99 ;
   min. monto:=9999;
   For i := 1 to TAM do begin
      suma:=suma+v[i].monto;
      calcularMinimo(v[i], min)
      vdias[v[i].dia]:=vdias[v[i].dia]+1;
  end;
  writeln ('El promedio de dinero transportado fue: ' ,suma/TAM );
  writeln ('El día que se transportó menos dinero fue : ' , diaMin, 'la distancia recorrida fue ', distancia);
  Viajespordia(vdias);
end;
```

Modularizar el cálculo de mínimo