

TEMA 2-CADP 2024 8/6//2024

Un fabricante de dispositivos electrónicos desea procesar información de repuestos que compró. El fabricante dispone de una estructura de datos con información de los 200 países de los cuales provienen repuestos, en la que para cada código de país (1..200) se tiene su nombre.

Realizar un programa que: a. Lea la información de los repuestos comprados y la almacene en una estructura de datos adecuada. De cada repuesto se lee código, precio y código del país del que proviene (entre 1 y 200). La lectura finaliza al ingresar el código -1 (que no debe procesarse).

b. Una vez cargada la información, procese la estructura de datos del inciso a e informe: 1. Cantidad de países para los que la cantidad de repuestos comprados es menor que el promedio de repuestos comprados a todos los países.

2. Para cada país, el nombre del país y el precio del repuesto más caro,

3. COMPLETO: La cantidad de repuestos que poseen al menos 3 ceros en su código.

Implemente el programa principal que invoque a los módulos de los incisos a y b. e imprima los resultados.

Resolución:

```
program parcial;
const
    DF=200;
    FIN=-1;
type
    rango=1..DF;
    cadena15=string[15];

    repuesto=record
        cod:integer;
        precio:real;
        codP:rango;
    end;
    lista=^nodo;
    nodo=record
        datos: repuesto;
        sig: lista;
    end;
```

```

vectorPaíses = array [rango] of string; //se dispone

vectorPrecios = array [rango] of real;

vectorCantRepuestos = array [rango] of integer;

//PROCESOS Y FUNCIONES

procedure CargarVector(var v:vectorPaíses); //se dispone

procedure LeerRepuesto(var r:repuesto);
begin
with r do
begin
writeln('Ingrese el codigo --FIN -1--');
readln(cod);
if ( cod <> FIN )then
begin
writeln('Ingrese el precio');
readln(precio);
writeln('Ingrese el codigo del pais --1 a 200');
readln(codP);
end;
end;
end;

procedure AgregarAdelante(var l:lista;r:repuesto);
var
nue:lista;
begin
new(nue);
nue^.datos:=r;
nue^.sig:=l;
l:=nue;
end;

procedure CargarLista(var l:lista);
var
r:repuesto;
begin
l:=nil;
LeerRepuesto(r);
while ( r.cod <> FIN )do
begin
AgregarAdelante(l,r);

```

```

        LeerRepuesto(r);
    end;
end;

procedure InicializarDatos(var v:vectorCantRepuestos; var
vec:vectorPrecios; var cantAL3Ceros:integer);
    var
        i:rango;
    begin
        for i:= 1 to DF do
            begin
                v[i]:=0;
                vec[i]:=-1;
            end;
            cantAL3Ceros:=0;
        end;

procedure actualizarMaximo(var maximo:real;nuevoValor:real);
    begin
        if ( nuevoValor > maximo) then maximo:=nuevoValor;
        end;

function AlMenosTresCeros (cod:integer):boolean;
    var
        cant:integer;
    begin
        cant:=0;
        while( (cod <> 0) and (cant >=3) )do
            begin
                if ( (cod mod 10) = 0 )then cant:=cant+1;
                cod:=cod div 10;
            end;
            AlMenosTresCeros:= (cant >= 3);
        end;

procedure Recorrer(l:lista; var v:vectorCantRepuestos; var
vec:vectorPrecios;var canTotal,cantAL3Ceros:integer);
begin
    canTotal:=0;
    InicializarDatos(v,vec,cantAL3Ceros);
    while ( l <> nil) do
        begin
            canTotal:=canTotal+1;
            v[l^.datos.codP]:=v[l^.datos.codP]+1;
            actualizarMaximo(vec[l^.datos.codP],l^.datos.precio);
            if ( SinCeros ( l^.datos.cod) )then cantSinCeros:=cantSinCeros+1;

```

```

        l:=l^.sig;
    end;
end;

procedure InformarMaximos(vec:vectorPrecios;v:vectorPaises);
var
    i:rango;
begin
    for i:= 1 to DF do writeln('Para el pai ', v[i], ' el precio mas
caro fue ', vec[i]);
end;

var
    l:lista;
    v:vectorPaises;
    vec:vectorPrecios;
    cantAL3Ceros:integer;
begin
    CargarVector(v); //se dispone
    CargarLista(l);
    Recorrer(l,vec,cantAL3Ceros);
    writeln(' ');
    InformarMaximos(vec,v);
    writeln('La cantidad de repuestos que poseen al menos 3 ceros en
su codigo son: ', cantAL3Ceros);
end.

```