Final Conceptos de Algoritmos Datos y Programas 11/04/2023

Apellido y Nombre Rodriguez Lucas

1. Un comercio dispone de una estructura de datos con las facturas (como máximo 2000 facturas) realizadas durante marzo de 2023. De cada factura se conoce el número de factura, código de cliente, código de sucursal y monto total. Las facturas se encuentran ordenadas por código de sucursal. Se pide implementar un programa con un módulo que reciba la estructura que se dispone y devuelva el código de sucursal con mayor cantidad de facturas. El programa debe informar el valor retornado por el módulo.

2. Dada la siguiente declaración y los siguientes procesos, indique para cada uno de los procesos si son correctos o no. El objetivo es duplicar el contenido del último nodo de la lista. Justifique su respuesta.

```
type miLista = ^nodo;
     nodo = record
              dato: integer; sig: miLista;
            end;
     lista = record
               pri: miLista; ult: miLista;
             end;
```

A

```
Procedure duplicar1 (L: lista);
begin
  L.ult^.dato:= L.ult^.dato * 2;
end;
```

B

```
Procedure duplicar2 (L: Lista);
var aux: miLista;
begin
  aux:= L.pri;
  while (aux^.sig <> nil) do
    aux:= aux^.sig;
  aux^.dato:= aux^.dato * 2;
end;
```

3. Calcule e indique la cantidad de **memoria estática y dinámica** que utiliza el siguiente programa. **Mostrar los valores intermedios para llegar al resultado y justificar.**

```
program ejercicio3;
const dimF = 200;
type cadena31 = string[31];
     alumno= record
               ape_nom: cadena31;
               promedio: real;
             end;
     vector = array [1..dimF] of ^alumno;
     lista = ^ nodo;
     nodo = record
          datos: alumno;
          sig:lista;
     end;
var  v: vector; a: alumno; nota, i, suma, cant: integer;
     aux: lista;
begin
  aux:= nil;
  for i:= 1 to dimF do
  begin
     read(a.ape_nom);read(nota); cant:= 0; suma:= 0;
     while (nota <> -1) do begin
       suma:= suma + nota; cant:= cant + 1;
       read(nota);
     end;
     if (cant <> 0) then a.promedio:= suma/cant
                    else a.promedio:= 0;
    new (v[i]);
    v[i]^:= a;
  end;
end.
```

Char	1 byte
Integer	4 bytes
Real	8 bytes
Boolean	1 byte
String	Longitud + 1 byte
Puntero	4 bytes

1.

```
program Final2023;

type
 factura = record
  numero:integer;
  cliente:integer;
  sucursal:integer;
  monto:real;
 end;

 vector = array of [1..2000] of factura;

 //procedure cargarVector (var v:vector;var dimL:integer); (se dispone y no se implementa)

 function sucursalMayorCantidad (v:vector;dimL:integer):integer;
 var
 sucursalActual:integer;
 cantActual:integer;
 cantMax:integer;
 sucursalMax:integer;
 i:integer;
 begin
 if (dimL>0) then
  begin
   sucursalActual:=v[1].sucursal; // si hay al menos un elemento inicializo
   cantMax:= -1;
   cantActual:=1;
   for i:=2 to dimL do //desde el segundo elemento (si hay) hasta dimL
    begin
     if (v[i].sucursal = sucursalActual) then // si estoy en la misma sucursal, sumo
      cantActual:=cantActual+1
     else
      begin
       if (cantActual > cantMax) then //sino verifico si debo actualizar el maximo
        begin
         cantMax:=cantActual;
         sucursalMax:=sucursalActual;
        end;
       sucursalActual:=v[i].sucursal; //reinicio las variables
       cantActual:=1;
      end;
   if (cantActual>cantMax) then //verifica la ultima sucursal.
    sucursalMax:=sucursalActual;
   sucursalMayorCantidad:=sucursalMax; //retorna
  end
 else
  sucursalMayorCantidad:=-1; // retorna -1 si no habia informacion en el vector
 end;

var
 v:vector;
 dimL:integer;
 max:integer;

begin
  cargarVector(v,dimL); // se dispone
  max:=sucursalMayorCantidad(v,dimL);
  if (max <> -1) then
   writeln('La sucursal con mayor cantidad de facturas es: ',max)
  else
   writeln ('No hay datos de facturas');
end.

```

2. Ambas están mal, porque ninguna verifica que la lista no este vacia. En A ult podría ser nil o no estar inicializado, y realizar esa operación daría un error en tiempo de ejecución.

B tampoco verifica que la lista no este vacia o no inicializada, si ese fuera el caso tendríamos un error en el while.

3.

Memoria estatica:
v:vector = 200*4 = 600

A:alumno = 40

Nota,i,suma,cant:integer = 4*4 =16

Aux:lista =4

Total = 600+40+16+4 = 660 bytes

Memoria dinámica:

For i:=1 to 200 // 200*40

Total = 200*40 = 8000 bytes