

# PARCIAL INTEGRADOR curso verano

Usted es miembro de un equipo de desarrollo. Otros integrantes del equipo diseñaron la estrategia de solución de un programa para el control de las ventas de una agencia de venta de televisores que comercializa 15 modelos diferentes con un plantel de 10 vendedores y que dispone de los siguientes datos:

VENDEDORES (máximo 10) almacenados en una estructura dinámica de PILA y sin orden con los siguientes datos: idVendedor (int), Apynombre (60)

MODELOS (máximo 15) con los siguientes datos: idModelo(int) descripcion (70) que deberán ser cargados por una función en la estructura que considere conveniente.

VENTAS.dat sin orden

idModelo	idVendedor	fecha	cliente
Int	Int	Ddmm	X(60)

Lo que se requiere es desarrollar un programa que permita emitir el siguiente listado ordenado por Modelo con el siguiente formato, agrupado por modelo y ordenado por idvendedor:

MODELO

Idvendedor nombreVendedor cliente fecha

Se pide:

- 1) Declarar todas las estructuras necesarias para resolver el problema (1 punto)
- 2) Declarar el prototipo de las funciones cargarModelos e imprimir que se detallan en los puntos 3 y 5. (1 punto)
- 3) Programar una función de carga de modelos (cargarModelos) con la firma que y estructura de datos que considere conveniente. (3 puntos)
- 4) Justifique la elección de la estructura del punto 3. (1 punto)
- 5) Programar una función que imprima el listado. (4 puntos)

Puede usar las funciones de biblioteca de la cátedra SIN DESARROLLAR en el examen,

```
//Se pide: desarrollar un programa que permita emitir el listado ordenado por Modelo.
//Agrupado por modelo, ordenado por idvendedor

1.
struct infovendedor
{int idvendedor;
 char apeynombre[60];}
infovendedores[10]

struct NodoPila
{ vendedor info;
  NodoPila* sgte=NULL;}

struct modelo
{ int idmodelo;
  char descripcion[70];}
modelo[15]

ventas.dat
struct ventas
{ int idModelo;
  int idVendedor;
  int fecha;
  char cliente[60];}
```

```

struct Nodoven
{
    ventas info;
    Nodoven* sgte=NULL;
}

2. CargarModelos (modelo v[], modelo info, int cant)
{
    modelo r;
    for (i=0; i<cant; i++)
    {
        cout << "Ingrese el id del modelo: " << endl;
        cin >> v[i].idmodelo;
        cout << "Ingrese la descripción del modelo: " << endl;
        cin >> v[i].descripcion;
    }
}

3. Como sabemos cuantos modelos tenemos se puede usar un array.

4.
vendedor pop (NodoPila* &pila);

void ventasalista (FILE *f, Nodoven* ven)
{
    ventas v;
    while (fread(&v, sizeof(ventas), 1, f))
    {
        insertarAlFinal (ven, v);
    }
}

//f=fopen("ventas.dat", "rb")
void emitirlistado (modelo v[], int cantmod, FILE *f, NodoPila* pila, infovendedor ven[], int cantven, Nodoven* ventas)
{
    int i, j;
    ventas v;
    Nodoven* ant= NULL;
    Nodoven* aux= NULL;
    for (i=0; i<cantven; i++)
    {
        ven[i]= pop(pila);
        ventasalista(f, ventas);
    }
    while ()
    {
        cout << "MODELO:" << j[i].idModelo;
        while (aux->sgte!=NULL)
        {
            if(ant->info.modelo==aux->info.idmodelo)
            {
                ant=aux;
                aux=aux->sgte;
            }
        }
    }
}

```