|  |
| --- |
| 2do Parcial Algoritmos y estructura de datos 2020 |
| Apellido y Nombre: Laureano Enrique Legajo: 1770640 |
| Debe desarrollar solo uno de los puntos según el valor del digito verificador del legajo  1,2 🡪 Punto 5;  3,4 🡪 Punto 4;  5,6 🡪 Punto 3;  7,8 🡪 Punto2;  9,0 🡪 Punto 1  Cualquier error o confusión que en la selección del punto a resolver invalida el examen |

Se quiere almacenar en memoria los gastos efectuados por cada uno de los 15 sectores de una empresa en un mes determinado para realizar un análisis estadístico.

Para ello se implementará una matriz de listas, donde las filas representan los sectores y las columnas los días y cada elemento es la lista con cada gasto de ese sector en ese día.

Los datos para armar la estructura se encuentran en el archivo de datos binario “Gastos.dat”, donde cada registro representa un gasto efectuado y tiene:

* Número de sector (de 1 a 15)
* Día del mes (de 1 a 31)
* Detalle
* Importe gastado.

El archivo no está ordenado y puede tener varios gastos realizados por un mismo sector.

1. Función **cargarEstructura** debe generar la estructura en memoria con los datos del archivo, tener en cuenta que la estructura hay que utilizarla en las otras funciones. Desarrollar todas las funciones utilizadas.
2. Función **mostrarGastos** debe mostrar por pantalla el siguiente listado:

Sector 1

Día 1

….. (detalle e importe de cada gasto)

…..

Día 2

…..

Sector 2

…..

Detalle e importe de cada gasto (ordenado de mayor a menor por importe) agrupado por sector y por día.

Desarrollar todas las funciones utilizadas.

1. Función **mostrarGastosSector** debe mostrar número de sector y sus gastos totales acumulados en el mes.
2. Función **mostrarGastos1Quincena** debe mostrar los datos acumulados en la primera quincena (del día 1 al día 14 incluidos)
3. Función **mayorGasto** debe mostrar qué sector y qué día se realizó el mayor gasto (suponer único)

#include <iostream>

#include<conio.h>

#include<cstring>

const int i=15;

const int k=31;

struct gasto{

int sector;

int diames;

char detalle[50];

int importe;

};

struct nodo{

gasto info;

nodo\* sgte;

};

nodo\* insertarlista(nodo\*&, gasto);

void cargarEstructura(FILE \*, nodo\* [i][k]);//al ser una matriz no es necesario pasar por referencia ya que se considera que los valores I y K van a cambiar dentro de la funcion

int main() {

FILE\* f;

nodo\* lista[i][k];//es una lista con filas de sectores y columnas de dias

//i representa sectores y k representa dias

for(int i=0;i<15;i++){

for(int k=0;k<31;k++){

lista[i][k]->sgte=NULL;//se colocan todos los punteros en null

}

}

cargarEstructura(f,lista); //llamamos a la funcion para que se carguen los datos de archivo en memoria

return 0;

}

void cargarEstructura(FILE \*f, nodo\* lista[i][k]){

gasto aux1;

f=fopen("Gastos.dat","rb+");

while(fread(&aux1,sizeof(gasto),1,f)){//vamos leyendo bloques del tamano gasto

//no requerimos de ningun contador, el mismo dato de int sector e int diames nos indica en que posicion de lista colocarlo

insertarlista(lista[aux1.sector-1][aux1.diames-1],aux1 );//se va guardando los datos obtenidos en los registros

//cuando se lee el bloque se inserta segun al sector o dia del mes que pertenece

//se le resta 1 para acomodar la diferencia dia 2, sector 14, iria en la posicion lista[13][1]

}

fclose(f);//cerramos el archivo

}

nodo\* insertarlista(nodo\*& p, gasto x){ //funcion para insertar informacion en la lista sin ningun criterio

nodo\* nuevo = new nodo();

nuevo->info=x;

nuevo->sgte=p;

p = nuevo;

return nuevo;

}