NOMBRE: Maximiliano Marcelo Meza

LEGAJO: 1757775

FECHA: 12/11/2020

Problema

Para una distribuidora mayorista de golosinas se debe desarrollar una aplicación que permita administrar

adecuadamente los productos que comercializa, organizados por rubros.

Dispone de los siguientes flujos de datos:

RUBROS; se trata de un archivo que contiene un registro por cada rubro con los siguientes datos (código de rubro

alfanumérico de 6 caracteres), descripción (alfabético de 50 caracteres), cantidad de artículos del rubro (numérico)

PRODUCTOS: se trata de un archivo que contiene un registro por cada producto con los siguientes datos (código de

producto (alfanumérico de 6 aracteres), descripción (alfabético de 50 caracteres), código de rubro (alfanumérico de 6

caracteres)

A partir de los archivos mencionados se deben generar una estructura en memoria que permita resolver recorrer

rápidamente los productos de cada rubro. Tener en cuenta que es importante conocer cuántos productos tiene cada

rubro sin necesidad de recorrer todos los productos del rubro

CODIGO C++:

#include<stdlib.h>

#include<stdio.h>

#include<ostream>

#include<fstream>

#include<iostream>

#include<string.h>

using namespace std;

typedef char str50[50];

typedef char str6[6];

struct rProducto{

str50 descripcion;

str6 codProducto;

str6 rubro;

};

struct rRubro{

str50 descripcion;

str6 codRubro;

int cantidad;

};

struct pProducto{

rProducto producto;

pProducto \*nodoSig;

};

struct pRubro {

rRubro rubro;

pRubro \*nodoSig;

pProducto \*nodoSL;

};

void inicializarProducto(rProducto &);

void inicializarRubro(rRubro &);

void cargarRubro (pRubro \*&);

void insertarRubroOrd(pRubro \*&, rRubro);

void cargarProductos(pRubro \*&);

void mostrarLista(pRubro \*);

pRubro \*buscarRubro(pRubro \*,str6);

void insertarOrdSLProd(pProducto \*&, rProducto);

void mostrarRubro(pRubro \*);

void mostrarSub(pProducto \*,bool&);

void contarCantidad(pRubro \*);

void agregarProducto(pRubro \*&);

int main(){

pRubro \*listaRubro;

listaRubro=NULL;

int op;

cargarRubro(listaRubro);

cargarProductos(listaRubro);

contarCantidad(listaRubro);

mostrarRubro(listaRubro) ;

agregarProducto(listaRubro) ;

system("pause");

return 0;

}

void agregarProducto(pRubro \*&lista){

pRubro \*aux;

aux=lista;

rProducto agregar;

str6 codRubro;

bool a=false;

system("cls");

inicializarProducto(agregar);

cout<<"Ingrese el codigo del rubro donde agregar producto: ";

cin.getline(codRubro,sizeof(str6));

cout<<"\nCodigo: ";

cin.getline(agregar.codProducto,sizeof(str6));

cout<<"\nDescripcion: ";

cin.getline(agregar.descripcion,sizeof(str50));

cout<<"\nRubro: ";

cin.getline(agregar.rubro,sizeof(str6));

while(lista!=NULL){

if (strcmp(lista->rubro.codRubro,codRubro)==0){

insertarOrdSLProd(lista->nodoSL,agregar);

mostrarSub(lista->nodoSL,a);

}

lista=lista->nodoSig;

}

system("pause");

}

void contarCantidad(pRubro \*lista){

pRubro \*aux;

pProducto \*subAux;

int cant=0;

aux=lista;

str6 clave;

strcpy(clave,"");

while(aux!=NULL){

strcpy(clave,aux->rubro.codRubro);

subAux=aux->nodoSL;

while(subAux!=NULL){

if(strcmp(subAux->producto.rubro,clave)==0){

cant++;

}

subAux=subAux->nodoSig;

}

aux->rubro.cantidad=cant;

cant=0;

aux=aux->nodoSig;

}

}

void mostrarRubro(pRubro \*lista){

pRubro \*pAuxRubro;

pAuxRubro=lista;

str6 codBuscar;

bool a=false;

system("cls");

cout<<"Ingrese el codigo del rubro a buscar: ";

cin.getline(codBuscar,sizeof(str6));

while ((pAuxRubro!=NULL)and (a==false)){

if (strcmp(pAuxRubro->rubro.codRubro,codBuscar)==0){

cout<<"Codigo de Rubro: "<<pAuxRubro->rubro.codRubro<<" Descripcion: "<<pAuxRubro->rubro.descripcion<<" Cantidad: "<<pAuxRubro->rubro.cantidad<<endl;

cout<<"----------------------->Productos<--------------------------\n";

mostrarSub(pAuxRubro->nodoSL,a);

}else{

pAuxRubro=pAuxRubro->nodoSig;

}

}

if (pAuxRubro==NULL){

cout<<"NO EXISTE TAL RUBRO."<<endl;

}

system("pause");

}

void mostrarSub(pProducto \*sub,bool &a){

pProducto \*aux;

aux=sub;

cout<<"CODIGO DESCRIPCION RUBRO\n";

while(aux!=NULL){

cout<<aux->producto.codProducto<<" "<<aux->producto.descripcion<<" "<<aux->producto.rubro<<endl;

aux=aux->nodoSig;

}

a=true;

}

void cargarProductos(pRubro \*&lista){

rProducto rAuxPro;

pRubro \*nodoLAux;

pProducto \*subLista;

inicializarProducto(rAuxPro);

nodoLAux=NULL;

subLista=NULL;

int primero=0,ultimo=0;

string cadena;

ifstream ar("Productos.txt");

while (!ar.eof()){

getline(ar,cadena);

ultimo=cadena.find(";",primero);

strcpy(rAuxPro.codProducto,cadena.substr(primero,ultimo-primero).c\_str());

primero=ultimo+1;

ultimo=cadena.find(";",primero);

strcpy(rAuxPro.descripcion,cadena.substr(primero,ultimo-primero).c\_str());

primero=ultimo+1;

ultimo=cadena.find(";",primero);

strcpy(rAuxPro.rubro,cadena.substr(primero,ultimo-primero).c\_str());

primero=ultimo+1;

nodoLAux=buscarRubro(lista,rAuxPro.rubro);

insertarOrdSLProd(nodoLAux->nodoSL,rAuxPro);

}

}

void insertarOrdSLProd(pProducto \*&sub, rProducto n){

pProducto \*nuevo\_nodo=new pProducto ();

pProducto \*aux1,\*aux2;

nuevo\_nodo->producto=n;

aux1=sub;

while ((aux1!=NULL)and (strcmp(aux1->producto.codProducto,n.codProducto)<0)){

aux2 = aux1;

aux1=aux1->nodoSig;

}

if (sub==aux1){

sub=nuevo\_nodo;

}else{

aux2->nodoSig=nuevo\_nodo;

}

nuevo\_nodo->nodoSig=aux1;

}

pRubro \*buscarRubro(pRubro \*lista,str6 codRubro){

pRubro \*aux;

aux=lista;

while ((aux!=NULL)){

if (strcmp(aux->rubro.codRubro,codRubro)==0){

return aux;

}

aux=aux->nodoSig;

}

return NULL;

}

void cargarRubro(pRubro \*&lista){

rRubro rAuxRubro;

int primero=0,ultimo=0;

string cadena;

inicializarRubro(rAuxRubro);

ifstream ar("Rubros.txt");

while (!ar.eof()){

getline(ar,cadena);

ultimo=cadena.find(";",primero);

strcpy(rAuxRubro.codRubro,cadena.substr(primero,ultimo-primero).c\_str());

primero=ultimo+1;

ultimo=cadena.find(";",primero);

strcpy(rAuxRubro.descripcion,cadena.substr(primero,ultimo-primero).c\_str());

primero=ultimo+1;

insertarRubroOrd(lista,rAuxRubro);

}

ar.close();

}

void insertarRubroOrd(pRubro \*&lista, rRubro n){

pRubro \*nuevo\_nodo=new pRubro ();

pRubro \*aux1,\*aux2;

nuevo\_nodo->rubro=n;

aux1=lista;

while ((aux1!=NULL)and ((strcmp(aux1->rubro.codRubro,n.codRubro))<0)){

aux2 = aux1;

aux1=aux1->nodoSig;

}

if (lista==aux1){

lista=nuevo\_nodo;

}else{

aux2->nodoSig=nuevo\_nodo;

}

nuevo\_nodo->nodoSig=aux1;

}

void inicializarRubro(rRubro &rubro){

strcpy(rubro.descripcion,"");

strcpy(rubro.codRubro,"");

rubro.cantidad=0;

}

void inicializarProducto(rProducto &producto){

strcpy(producto.descripcion,"");

strcpy(producto.codProducto,"");

strcpy(producto.rubro,"");