**Solucionario del Examen Parcial de SI205**

/\*Pregunta 1.

Para un grupo de n (n<=50) productos se desea almacenar la siguiente información código (entero de tres cifras), precio (real entre 1.00 y 9.90 soles) y tipo de producto (A, B, C) luego se pide:

a) Ordenar los datos por Tipo de Producto en ascendente y precio en descendente y mostrarlos

b) Eliminar los productos del tipo "B" cuyo precio sea menor al promedio de precios del tipo B

c) Ingresar un nuevo producto e insertarlo en la lista de datos de tal forma que se mantenga el orden por tipo de producto y precio

d) Mostrar los datos resultantes

\*/

#include <iostream>

using namespace std;

//Estructura

struct Reg{

int Cod;

float precio;

char tipo;

};

int main(){

int Cod;

float precio,prom,s;

char tipo;

int n,i,j,num1,num2,cont;

Reg aux;

bool completo,existe;

//Ingreso de datos

do{

cout<<"Ingrese # de productos:"; cin>>n;

}while(!(n>0));

//Creando el arreglo de estructuras

Reg p[n];

for(i=0; i<n; i++){

cout<<"Producto #"<<(i+1)<<endl;

do{

cout<<"Ingrese codigo:"; cin>>Cod;

}while(!(100<=Cod and Cod<=999));

p[i].Cod=Cod;

do{

cout<<"Ingrese precio:( entre 1,00 y 9,90):"; cin>>precio;

}while(!(1<=precio and precio<=9.9));

p[i].precio=precio;

do{

cout<<"Ingrese tipo (A, B, C):"; cin>>tipo;

tipo=toupper(tipo);

}while(!('A'<=tipo and tipo<='C'));

p[i].tipo=tipo;

}

cout<<"Datos sin ordenar..."<<endl;

cout<<"Codigo\tPrecio\tTipo"<<endl;

for(i=0; i<n; i++){

cout<<p[i].Cod<<"\t"<<p[i].precio<<"\t"<<p[i].tipo<<endl;

}

//Parte a)

cout<<"Datos ordenados..."<<endl;

for(i=0; i<n-1; i++){

for(j=i+1; j<n; j++){

//Para ordenar en forma ascendente respecto al tipo y descendente respecto al precio

num1=p[i].tipo\*1000-p[i].precio;

num2=p[j].tipo\*1000-p[j].precio;

if(num1>num2){

aux=p[i];

p[i]=p[j];

p[j]=aux;

}

}

}

cout<<"Datos ordenados..."<<endl;

cout<<"Codigo\tPrecio\tTipo"<<endl;

for(i=0; i<n; i++){

cout<<p[i].Cod<<"\t"<<p[i].precio<<"\t"<<p[i].tipo<<endl;

}

//Parte b)

s=0; cont=0;

for(i=0; i<n; i++){

if(p[i].tipo=='B'){

s=s+p[i].precio;

cont++;

}

}

if(cont>0){

prom=s/cont;

}

else{

prom=0;

}

i=0;

while(i<n){

if(p[i].precio<prom and p[i].tipo=='B'){

j=i;

while(j<n-1){

p[j]=p[j+1];

j++;

}

n--;

}

else{

i++;

}

}

cout<<"Datos despues de la eliminacion..."<<endl;

cout<<"Codigo\tPrecio\tTipo"<<endl;

for(i=0; i<n; i++){

cout<<p[i].Cod<<"\t"<<p[i].precio<<"\t"<<p[i].tipo<<endl;

}

//Parte c)

cout<<"Nuevo producto:"<<endl;

do{

cout<<"Ingrese codigo:"; cin>>Cod;

}while(!(100<=Cod and Cod<=999));

do{

cout<<"Ingrese precio:( entre 1,00 y 9,90):"; cin>>precio;

}while(!(1<=precio and precio<=9.9));

do{

cout<<"Ingrese tipo (A, B, C):"; cin>>tipo;

tipo=toupper(tipo);

}while(!('A'<=tipo and tipo<='C'));

i=0;

num2=tipo\*1000-precio;

existe=false;

completo=false;

while(i<n and !completo){

num1=p[i].tipo\*1000-p[i].precio;

if(num2>num1){

i++;

}

else{

completo=true;

if(num2==num1){

existe=true;

}

}

}

if(!existe){

j=n-1;

while(j>=i){

p[j+1]=p[j];

j--;

}

p[i].Cod=Cod;

p[i].precio=precio;

p[i].tipo=tipo;

n++;

}

else{

cout<<"Producto ya existe..."<<endl;

}

cout<<"Datos despues de la insercion..."<<endl;

cout<<"Codigo\tPrecio\tTipo"<<endl;

for(i=0; i<n; i++){

cout<<p[i].Cod<<"\t"<<p[i].precio<<"\t"<<p[i].tipo<<endl;

}

return 0;

}

/\*Pregunta 2:

Una Universidad tiene n Facultades representadas por los números del 1 a n. Cada Facultad cuenta con un número de estudiantes (máximo 100). Cada estudiante tiene un año de ingreso. Elabore un programa que:

a) Permita almacenar el año de ingreso de cada estudiante de cada Facultad.

b) Determine por cada Facultad el número de estudiantes por año de ingreso.

\*/

#include<iostream>

#include<string>

#include<stdlib.h>

#include<time.h>

using namespace std;

int main() {

}

/\*Pregunta 3:

Escriba un programa que permita ingresar un número entero N positivo mayor de tres cifras y presente el número modificado de tal forma que no contenga la cifra o cifras con el mayor número de apariciones (el mayor número de apariciones de ser mayor a uno). Ejemplo:

Si ingresa el número: 164546964

El mayor número de apariciones es: 3

Los dígitos con el mayor número de apariciones son: 6 y 4

El número modificado será: 159.

\*/

#include <iostream>

using namespace std;

int main(){

int c,N,aux,r,i,j,k,max=0;

do{

cout<<"Ingrese el numero N mayor de 3 cifras: "; cin>>N;

}while(!(N>=1000));

//Numero de cifras

c=0;

aux=N;

while(aux!=0){

aux/=10;

c++;

}

//Contar el numero de apariciones de cifras y meter al numero en un arreglo

int Cif[10], Num[c];

for(i=0; i<=9; i++){

Cif[i]=0;

}

aux=N;

i=0;

while(aux!=0){

r=aux%10;

Num[i]=r;

i++;

aux/=10;

Cif[r]++;

}

//Hallando el mayor numero de apariciones

for(i=0; i<=9; i++){

if(max<Cif[i]){

max=Cif[i];

}

}

if(max>1){

cout<<"\nEl mayor numero de apariciones es: "<<max<<endl;

//Eliminando los numeros con el mayor numero de apariciones

cout<<"\nEl digito o los digitos con el mayor numero de apariciones: ";

for(i=0; i<=9; i++){

if(Cif[i]==max){

cout<<i<<" ";

j=0;

while(j<c){

if(i==Num[j]){

k=j;

while(k<c-1){

Num[k]=Num[k+1];

k++;

}

c--;

}

else{

j++;

}

}

}

}

//Presentacion

N=0;

for(i=c-1; i>=0; i--){

N=10\*N+Num[i];

}

cout<<"\n\nNumero modificado: "<<N;

}

else{

cout<<"\nNo hay digito que tenga mayor numero de apariciones que los demas"<<endl;

}

return 0;

}