parcial.cpp

```
/*Andrey Felipe Orozco Montoya 20221578022
2 Modelar dos arreglos dinámicos (int, float) siguiendo las siguientes reglas:*/
#include

using namespace std;

class sobrecarga {//Declaracion de la clase
private:
int ocupacion;/*Definicion de variable global que funciona como acumulador*/

public://Definicion de metodos de la clase
sobrecarga(float*,int*,int);
int getOcupacion(void);
void operator+(float*);
void operator+(int*);
void operator-(float*);
void operator-(int*);
void operator*(float*);
void operator*(int*);
int menu();
};

//Definicion de metodos de la clase

sobrecarga::sobrecarga(float *flotantes,int *enteros,int tamano) {/*Constructor de
la clase usado para inicializar los vectores y la variable de la clase*/
ocupacion=0;
for(int i=0; i flotantes[i]=0.0;
enteros[i]=0;
}
}

int sobrecarga::getOcupacion() {/*Metodo encargado de retornar el valor de la
variable de la clase*/
return ocupacion;
}

int sobrecarga::menu() {/*Metodo encargado de mostrar un menu y recibir una opcion
de este mismo*/
int opcion=0;
cout<<"------------------MENU--------------------"< cout<<"1. Insertar"< cout<<"2.
Eliminar"< cout<<"3. Imprimir"< cout<<"4. Salir"< cout<<"Por favor ingrese una
opción"< cin>>opcion;
if(opcion<1 || opcion>4) {//Validacion de opcion ingresada
cout<<"Opcion invalida"< }
return opcion;
}
```

```
void sobrecarga::operator+(float*flotantes) {/*Metodo con sobrecarga de operador y
metodo para recibir el valor de flotantes*/
float valor=0.0;
cout<<"Por favor ingrese el valor del vector flotante"< cin>>valor;
flotantes[ocupacion]=valor;
cout<}

void sobrecarga::operator+(int*enteros) {/*Metodo con sobrecarga de operador y
metodo para recibir el valor de enteros*/
int valor=0;
cout<<"Por favor ingrese el valor del vector entero"< cin>>valor;
enteros[ocupacion]=valor;
ocupacion=ocupacion+1;
system("pause");
system("cls");
}

void sobrecarga::operator-(float*flotantes) {/*Metodo con sobrecarga de operador y
metodo para eliminar la cabeza del vector flotante*/
for(int i=0; i flotantes[i]=flotantes[i+1];
}
cout<<"Flotante eliminado correctamente"<}

void sobrecarga::operator-(int*enteros) {/*Metodo con sobrecarga de operador y
metodo para eliminar la cabeza del vector entero*/
for(int i=0; i enteros[i]=enteros[i+1];
}
ocupacion=ocupacion-1;
cout<<"Entero eliminado correctamente"< system("pause");
system("cls");
}

void sobrecarga::operator*(float*flotantes) {/*Metodo con sobrecarga de operador y
metodo para mostrar en pantalla el vector flotante*/
cout<<"ARREGLO FLOTANTE"< for(int i=0; i cout< }
cout<}

void sobrecarga::operator*(int*enteros) {/*Metodo con sobrecarga de operador y
metodo para mostrar en pantalla el vector*/
cout<<"ARREGLO ENTEROS"< for(int i=0; i cout< }
cout< system("pause");
system("cls");
}

int main() {
int *enteros;/*Definicion de direcciones de memorias para asignacion dinamica de
memoria*/
float *flotantes;
int tamano=0;//Variable que contendra el tamaño de los vectores
while(tamano<=0) {//Ciclo para validar un tamaño plausible de los vectores
cout<<"Por favor ingrese el tamano de los vectores "< cin>>tamano;
```

```
}
//Asignacion de memoria dinamica de los vectores
enteros=new int[tamano];
flotantes=new float[tamano];
if(flotantes==NULL || enteros==NULL) {//Validacion de la asignacion dinamica de
memoria
cout<<"Error en asignacion dinamica de memoria"< return 42;
}
sobrecarga a(flotantes,enteros,tamano);/*Declaracion de un objeto de clase
"Sobrecarga " y envio de vectores para inicializar en el constructor*/

int opcion=0;/*Variable que maneja la opcion ingresada al switch case*/

while(opcion!=4) {/*Ciclo que repite el menu hasta que se seleccione salir*/

opcion=a.menu();/*Asignacion a la variable opcion por medio del metodo "menu"*/

switch(opcion) {
case 1://Agregar
if(a.getOcupacion()!=tamano) {/*Validacion de que el vector no se encuentre
lleno*/
a+flotantes;
a+enteros;
} else {
cout<<"Los vectores se encuentran llenos."< }
break;
case 2://ELiminar
if(a.getOcupacion()!=0) {/*Validacion para que el vector no se encuentre vacio*/
a-flotantes;
a-enteros;
} else {
cout<<"Los vectores se encuentran vacios, no se podra eliminar nada."< }
break;
case 3://Imprimir
if(a.getOcupacion()!=0) {/*Validacion para que el vector no se encuentre vacio*/
a*flotantes;
a*enteros;
} else {
cout<<"Los vectores se encuentran vacios, no se podra visualizar nada."< }
break;
}
}
cout<<"Gracias por usar el programa."< //Liberacion de memoria dinamica
delete enteros;
delete flotantes;
return 0;
}
```