



UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

Estructuras computacionales

Ingeniería en computación inteligente

Unidad 2 Listas circulares dobles y N problemas con diferentes tipos de lista
(Parcial 2)

Nombre del maestro: Miguel Ángel Meza de Luna.

Nombre de los alumnos:

Ángel David Ortiz Quiroz ID 261481

Ximena Rivera Delgadillo ID 261261

Erick Iván Ramírez Reyes ID 260806

Diego Emanuel Saucedo Ortega ID 261230

Jose Luis Sandoval Perez ID 261731

Carlos Daniel Torres Macias ID 244543

Fecha de entrega: domingo 27 de Marzo del 2021.

CODIGO LISTAS SIMPLES

(Nivel de entendimiento básico)

```
//PROYECTO 2 parcial II
//PROGRAMA II
//Estructuras Computacionales 2°A (PARCIAL 2)
/*
Ángel David Ortiz Quiroz          ID: 261481
Ximena Rivera Delgadillo          ID: 261261
Erick Ivan Ramirez Reyes          ID: 260806
Diego Emanuel Saucedo Ortega      ID: 261230
Jose Luis Sandoval Perez          ID: 261731
Carlos Daniel Torres Macias       ID: 244543
*/

#include<iostream>
#include<stdlib.h>
using namespace std;

struct node {
    int dato;
    struct node* sig;
};

struct node* inicio, * tmp;
```

```

int main() {

    int numnod = 0, op = 0;

    cout << "El siguiente programa nos permitirá conocer el
funcionamiento de listas simples" << endl;

    cout << "permitiendo crear nodos infinitos en una lista asi
como recorrer la lista creada" << endl;

    cout << "-----
-----\n";

    system("pause");
    system("cls");

    while (op != 5) {
        cout << "\tM E N U" << endl;
        cout << "1. Agregar nodos a la LISTA" << endl;
        cout << "2. Recorrer nodos de la LISTA" << endl;
        cout << "3. Buscar un nodo en la LISTA" << endl;
        cout << "4. Eliminar un nodo en l LISTA" << endl;
        cout << "5. Salir" << endl;
        cout << "Ingresa una opcion: "; cin >> op;
        cout << "-----
-----\n";

        system("pause");
        system("cls");
        if (op == 1) {
            cout << "A G R E G A R nodos a la L I S T A" << endl;
            if (numnod == 0) {
                inicio = new struct node();
                cout << "Ingresa el valor que deseas que tenga el
nodo en la lista: "; cin >> inicio->dato;

```

```

        inicio->sig = NULL;
        numnod++;
    }
    else {
        tmp = new node();
        cout << "Ingresa el valor que deseas que tenga el
nodo en la lista: "; cin >> tmp->dato;
        tmp->sig = inicio;
        inicio = tmp;
        numnod++;
    }
    cout << "-----
-----\n";

    system("pause");
    system("cls");
}

if (op == 2) {
    cout << "R E C O R R E R nodos de la lista" << endl;
    if (numnod == 0) {
        cout << "ERROR, no hay nodos en la lista" << endl;
    }
    else {
        tmp = inicio;
        while (tmp != NULL) {
            cout << tmp->dato;
            cout << endl;
            tmp = tmp->sig;
        }
    }
}

```

```

        cout << "-----\n";

        system("pause");
        system("cls");
    }
    if (op == 3) {
        cout << "B U S C A R un nodo en la lista" << endl;
        tmp = inicio;
        int value;
        if(numnod==0){
            cout << "ERROR, no hay nodos en la lista" << endl;
        }
        else{
            cout << "Ingresa el nodo que deseas buscar: "; cin
>>value;

            if (tmp->dato!= value && tmp!= NULL)
            {
                cout << "No existe el nodo "<< value << " en la
lista" << endl;

            }
            else {
                cout << "El nodo "<< value << " si se
encuentra en la lista" << endl;
            }
        }

        cout << "-----\n";

        system("pause");
        system("cls");
    }
}

```

```

    }
    if (op == 4) {
        cout << "E L I M I N A R un nodo en la lista" << endl;
        tmp = inicio;
        int value;

        if(numnod==0){
            cout << "ERROR, no hay nodos en la lista" << endl;
        }
        else{
            cout << "Ingresa el nodo que deseas eliminar: ";
cin >> value;
            if (tmp->dato != value && tmp != NULL)
            {
                while (tmp->sig->dato != value && tmp->sig !=
NULL)

                {
                    tmp = tmp->sig;
                }

                if (tmp->sig == NULL)
                {
                    cout << "No existe el nodo " << value << "
en la lista" << endl;
                }
                else
                {
                    node* tmp2, * tmp3;
                    tmp2 = tmp->sig;
                    tmp3 = tmp->sig->sig;

```

```

        tmp->sig = tmp3;
        delete(tmp2);
        cout << "El nodo " << value << " ha sido
eliminado de la lista" << endl;
        numnod--;
    }
}
else
{
    inicio = inicio->sig;
    delete(tmp);
    cout << "El nodo " << value << " ha sido
eliminado de la lista" << endl;
    numnod--;
}
}

    cout << "-----\n";
    system("pause");
    system("cls");
}
}

    cout << "-----\n";

    cout << "PROGRAMA HECHO POR:" << endl;
    cout << "Angel David Ortiz Quiroz          ID : 261481" << endl;
    cout << "Ximena Rivera Delgadillo          ID : 261261" << endl;
    cout << "Erick Ivan Ramirez Reyes          ID : 260806" << endl;
    cout << "Diego Emanuel Saucedo Ortega        ID : 261230" << endl;

```

```
cout << "Jose Luis Sandoval Perez      ID : 261731" << endl;
cout << "Carlos Daniel Torres Macias   ID : 244543" << endl;
system("pause");
system("cls");
return 0;
}
```


CODIGO LISTAS SIMPLES 2

(Nivel de entendimiento básico)

```
//PROYECTO 2 parcial II
```

```
//PROGRAMA II
```

```
//Estructuras Computacionales 2Â°A (PARCIAL 2)
```

```
/*
```

```
Ãngel David Ortiz Quiroz          ID: 261481
```

```
Ximena Rivera Delgadillo          ID: 261261
```

```
Erick Ivan Ramirez Reyes          ID: 260806
```

```
Diego Emanuel Saucedo Ortega      ID: 261230
```

```
Jose Luis Sandoval Perez          ID: 261731
```

```
Carlos Daniel Torres Macias       ID: 244543
```

```
*/
```

```
#include<iostream>
```

```
#include<stdlib.h>
```

```
using namespace std;
```

```
struct node {
```

```
    string name;
```

```
    struct node* sig;
```

```
};
```

```
struct node* inicio, * tmp;
```

```

int main() {

    int numnod = 0;

    cout << "El siguiente programa nos permitira conocer el
funcionamiento de listas simples" << endl;

    cout << "permitiendo crear nodos infinitos en una lista asi
como recorrer la lista creada" << endl;

    cout << "Este programa simulara una base de datos de nombres
de trabajadores" << endl;

    cout << "-----\n";

    system("pause");
    system("cls");
    int op=0;
    while (op != 4) {
        op=0;
        cout << "\tM E N U" << endl;
        cout << "1. Agregar Trabajadores a la LISTA" << endl;
        cout << "2. Desplegar trabajadores de la LISTA" << endl;
        cout << "3. Eliminar un trabajador en l LISTA" << endl;
        cout << "4. Salir" << endl;
        cout << "Ingresa una opcion: ";
        cin >> op;
        cout << "-----\n";

        system("pause");
        system("cls");
    }
}

```

```

        if (op == 1) {
            cout << "A G R E G A R trabajadores a la L I S T A" <<
endl;

            if (numnod == 0) {
                inicio = new struct node();

                cout << "Ingresa el nombre del trabajador que
deseas que tenga el en la lista(nombre seguido de un _ para
ingresar apellido): "; cin >> inicio->name;

                inicio->sig = NULL;
                numnod++;
            }
            else {
                tmp = new node();

                cout << "Ingresa el nombre del trabajador que
deseas que tenga el en la lista(nombre seguido de un _ para
ingresar apellido): "; cin >> tmp->name;

                tmp->sig = inicio;
                inicio = tmp;
                numnod++;
            }

            cout << "-----
-----\n";

            system("pause");
            system("cls");
        }

        if (op == 2) {
            cout << "D E S P L E G A R  trabajadores de la lista"
<< endl;

            if (numnod == 0) {
                cout << "ERROR, no hay trabajadores en la lista"
<< endl;

```

```

    }
    else {
        tmp = inicio;
        while (tmp != NULL) {
            cout << tmp->name;
            cout << endl;
            tmp = tmp->sig;
        }
    }

    cout << "-----\n";

    system("pause");
    system("cls");
}

if (op == 3) {
    cout << "E L I M I N A R un trabajador en la lista" <<
endl;

    cout << "-----\n";

    if (numnod == 0) {
        cout << "ERROR, no hay trabajadores en la lista"
<< endl;
    }
    else{
        tmp = inicio;
        cout << "L I S T A E X I S T E N T E " << endl;
        for(int i=0; i<numnod;i++){

            while (tmp != NULL) {
                cout << tmp->name;

```

```

        cout << endl;
        tmp = tmp->sig;
    }
}

string deletedname;
cout << "Ingresa el nombre del trabajador que deseas
eliminar: "; cin >> deletedname;
if (tmp->name!= deletedname && tmp != NULL)
{
    while (tmp->sig->name != deletedname && tmp->sig
!= NULL)
    {
        tmp = tmp->sig;
    }

    if (tmp->sig == NULL)
    {
        cout << "No existe el trabajador " <<
deletedname << " en la lista" << endl;
    }
    else
    {
        node* tmp2, * tmp3;
        tmp2 = tmp->sig;
        tmp3 = tmp->sig->sig;

        tmp->sig = tmp3;
        delete(tmp2);
    }
}

```

```

        cout << "El trabajador " << deletedname << "
ha sido eliminado de la lista" << endl;
        numnod--;
    }
}
else
{
    inicio = inicio->sig;
    delete(tmp);
    cout << "El trabajador " << deletedname << " ha
sido eliminado de la lista" << endl;
    numnod--;
}
cout << "-----
-----\n";

system("pause");
system("cls");
}
system("pause");
system("cls");
}
}

cout << "-----
-----\n";

cout << "PROGRAMA HECHO POR:" << endl;
cout << "Angel David Ortiz Quiroz          ID : 261481" << endl;
cout << "Ximena Rivera Delgadillo          ID : 261261" << endl;
cout << "Erick Ivan Ramirez Reyes          ID : 260806" << endl;
cout << "Diego Emanuel Saucedo Ortega        ID : 261230" << endl;
cout << "Jose Luis Sandoval Perez            ID : 261731" << endl;

```

```
    cout << "Carlos Daniel Torres Macias      ID : 244543" << endl;  
    system("pause");  
    system("cls");  
    return 0;  
}
```

CODIGO LISTAS CIRCULAS SIMPLE (NIVEL DE ENTENDIMIENTO BASICO- INTERMEDIO)

```
//LISTA CIRCULAR SIMPLE
//PROYECTO 2 parcial II
//PROGRAMA II
//Estructuras Computacionales 2°A (PARCIAL 2)
/*
Ángel David Ortiz Quiroz          ID: 261481
Ximena Rivera Delgadillo          ID: 261261
Erick Ivan Ramirez Reyes          ID: 260806
Diego Emanuel Saucedo Ortega      ID: 261230
Jose Luis Sandoval Perez          ID: 261731
Carlos Daniel Torres Macias       ID: 244543
*/
#include <stdlib.h>
#include <iostream>
#include <stdio.h>
using namespace std;

struct Nodo{
    int dato;
    Nodo *sig;
}*inicio,*fin;

void insertar(){
    if(inicio==NULL){
```



```

        inicio = new Nodo();
        cout<<"Ingresa el nuevo dato: ";
        cin >> (inicio->dato);
        inicio->sig=inicio;
    }//el nodo esta vacio
    else if(fin==NULL){
        fin = new Nodo();
        cout<<"Ingresa el nuevo dato: ";
        cin >> (fin->dato);
        fin->sig=inicio;
        inicio->sig=fin;
    }
    else{
        Nodo *nuevo= new Nodo();
        cout<<"Ingresa el nuevo dato: ";
        cin >> (nuevo->dato);
        fin->sig=nuevo;
        nuevo->sig=inicio;
        fin=nuevo;
    }

    cout<<"\n Nodo insertado completamente\n\n";
}

void mostrar(){
    Nodo *actual=new Nodo();
    actual=inicio;
    if (inicio!=NULL){
        int op;

```

```

do{
    cout<<"\nLISTA CIRCULAR SIMPLE\n\n";
    cout<<"El dato actual es: "<<actual->dato<<endl;
    cout << "-----\n";
    -----\n";

    cout<<"1.Mostrar dato siguiente\n";
    cout<<"2.Salir\nOpcion: ";
    cin>>op;
    switch(op){
        case 1:
            actual=actual->sig;
            break;
        case 2:
            break;
        default:
            cout<<"No es una opcion valida\n";
            break;
    }//fin switch

    system("pause");
    system("cls");
}while(op!=2);
}
else{
    cout<<"\n No hay elementos en la lista \n\n";
}
}

void eliminar(){

```

```
    if(inicio==NULL){cout<<"No hay elementos por borrar\n";  
return;}
```

```
    Nodo *actual= new Nodo();
```

```
    if(inicio->sig==fin){
```

```
        inicio->sig=inicio;
```

```
        actual=fin;
```

```
        delete actual;
```

```
        fin=NULL;
```

```
    }
```

```
    else if(inicio->sig==inicio){
```

```
        inicio=NULL;
```

```
    }
```

```
    else{
```

```
        actual=inicio;
```

```
        fin->sig=inicio->sig;
```

```
        inicio=inicio->sig;
```

```
        delete actual;
```

```
    }
```

```
}
```

```
int main(){
```

```
    int op,nodos;
```

```
    inicio=NULL;
```

```
    fin=NULL;
```

```
    do{
```

```
        cout<<"1.Insertar nodo"<<endl;
```

```
        cout<<"2.Mostrar lista"<<endl;
```

```
        cout<<"3.Eliminar inicio"<<endl;
```

```

cout<<"4.Salir "<<endl;
cout<<"Opcion: ";
cin>>op;
switch (op){
    case 1:
        insertar();
        break;
    case 2:
        mostrar();
        break;
    case 3:
        eliminar();
        break;
    case 4:
        cout<<"Poder decir adios, es crecer...\n";
        cout << "-----\n";
        break;
    default:
        cout<<"No es una opcion\n";
        cout << "-----\n";
        break;
}

}

system("pause");
system("cls");
}while(op!=4);

cout << "PROGRAMA HECHO POR:" << endl;

```

```
    cout << "Angel David Ortiz Quiroz          ID : 261481" << endl;
    cout << "Ximena Rivera Delgadillo          ID : 261261" << endl;
    cout << "Erick Ivan Ramirez Reyes          ID : 260806" << endl;
    cout << "Diego Emanuel Saucedo Ortega        ID : 261230" << endl;
    cout << "Jose Luis Sandoval Perez            ID : 261731" << endl;
    cout << "Carlos Daniel Torres Macias          ID : 244543" << endl;
    cout << "-----\n";
    system("pause");
    return 0;
}
```

CODIGO LISTAS CIRCULAR DOBLEMENTE ENLAZADA (NIVEL DE ENTENDIMIENTO INTERMEDIO)

```
//PROYECTO 2 parcial II
//PROGRAMA LISTAS CIRCULAR DOBLEMENTE ENLAZADAS
//Estructuras Computacionales 2ºA (PARCIAL 2)
/*
Ángel David Ortiz Quiroz          ID: 261481
Ximena Rivera Delgadillo          ID: 261261
Erick Ivan Ramirez Reyes          ID: 260806
Diego Emanuel Saucedo Ortega      ID: 261230
Jose Luis Sandoval Perez          ID: 261731
Carlos Daniel Torres Macias       ID: 244543
*/
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
using namespace std;

struct nodo{
    int dato;
    nodo *sig;
    nodo *ant;
};

void insertarDespues();
```

```

void insertarAntes();
void eliminarNodo();
void mostrarLista();

nodo *inicio,*fin,*tmp;

int main(){

    inicio=NULL;
    fin=NULL;

    int op=-1;
    do{
        cout << "\tM E N U" << endl;
        cout<<"1.Insertar dato\n";
        cout<<"2.Borrar dato\n";
        cout<<"3.Mostrar lista\n";
        cout<<"4.Salir\nSeleccion--->";
        cout << "Ingresa una opcion: "; cin >> op;
        cout << "-----\n";
        -----\n";

        system("pause");
        system("cls");

        if(op==1){
            if(inicio==NULL){
                inicio=new nodo();
                cout<<"Ingresa nuevo dato inicio--->";
                cin>>inicio->dato;
                inicio->ant=inicio;
            }
        }
    }while(op!=4);
}

```

```

        inicio->sig=inicio;
    }//no hay elementos en la pila

else if(inicio->sig==inicio && inicio->ant==inicio){
    fin=new nodo();
    cout<<"Ingrese nuevo dato final--->";
    cin>>fin->dato;
    inicio->sig=fin;
    inicio->ant=fin;
    fin->sig=inicio;
    fin->ant=inicio;
}//solo hay un elemento, se crea nodo de fin

else{
    int opp=-1;
    tmp=inicio;
    do{
        cout<<"Dato Actual: "<<tmp->dato<<endl;
        cout<<"1.Mostrar dato siguiente\n";
        cout<<"2.Mostrar dato anterior\n";
        cout<<"3.Insertar nuevo dato antes del
actual\n";

        cout<<"4.Insertar nuevo dato despues del
actual\n";

        cout<<"5.Salir\nSeleccion--->";
        cout << "-----\n";
        -----\n";

        system("pause");
        system("cls");
        cout << "Ingresa una opcion: "; cin>>opp;

```



```

        if(opp==1){tmp=tmp->sig;}
        if(opp==2){tmp=tmp->ant;}
        if(opp==3){insertarAntes();}
        if(opp==4){insertarDespues();}

        system("pause");
        system("cls");
    }while(opp!=5);
} //hay elementos en la pila

} //fin insertar dato

if(op==2){
    if(inicio==NULL){cout<<"No hay elementos que
eliminar.\n";}
    else{eliminarNodo();} //si hay elementos
} //eliminar nodo

if(op==3){
    if(inicio==NULL){cout<<"No hay elementos en la
lista.\n";}
    else{mostrarLista();} //si hay elementos
} //mostrar datos
cout << "-----\n";
-----\n";

    system("pause");
    system("cls");
}while(op!=4);
cout << "-----\n";
-----\n";

cout << "PROGRAMA HECHO POR:" << endl;

```

```

        cout << "Angel David Ortiz Quiroz          ID : 261481" << endl;
        cout << "Ximena Rivera Delgadillo          ID : 261261" << endl;
        cout << "Erick Ivan Ramirez Reyes          ID : 260806" << endl;
        cout << "Diego Emanuel Saucedo Ortega        ID : 261230" << endl;
        cout << "Jose Luis Sandoval Perez            ID : 261731" << endl;
        cout << "Carlos Daniel Torres Macias          ID : 244543" << endl;
        system("pause");
        system("cls");
return 0;
}

```

```

void insertarDespues(){
    if(tmp==fin){
        nodo *aux=new nodo();
        cout<<"Ingrese dato despues del final--->";
        cin>>aux->dato;
        aux->ant=fin;
        aux->sig=inicio;
        fin->sig=aux;
        inicio->ant=aux;
        fin=aux;
    }//es el ultimo elemento
    else{
        nodo *aux=new nodo();
        cout<<"Ingrese dato despues--->";
        cin>>aux->dato;
        aux->ant=tmp;
        aux->sig=tmp->sig;
        tmp->sig->ant=aux;
    }
}

```

```

        tmp->sig=aux;
    }
}

void insertarAntes(){
    if(tmp==inicio){
        nodo *aux=new nodo();
        cout<<"Ingrese dato antes del inicio--->";
        cin>>aux->dato;
        aux->ant=fin;
        aux->sig=inicio;
        inicio->ant=aux;
        fin->sig=aux;
        inicio=aux;
    }//es el primer elemento
    else{
        nodo *aux=new nodo();
        cout<<"Ingrese dato antes--->";
        cin>>aux->dato;
        aux->ant=tmp->ant;
        aux->sig=tmp;
        tmp->ant->sig=aux;
        tmp->ant=aux;
    }
}

```

```

void eliminarNodo(){
    int op=-1;
    tmp=inicio;

```

```

do{
    cout<<"Dato Actual: "<<tmp->dato<<endl;
    cout<<"1.Mostrar dato siguiente\n";
    cout<<"2.Mostrar dato anterior\n";
    cout<<"3.Eliminar dato actual\n4.Salir\n";
    cout<<"Seleccion--->";
    cin>>op;
    if(op==1){tmp=tmp->sig;}
    if(op==2){tmp=tmp->ant;}
    if(op==3){
        if(inicio->sig==inicio && inicio->ant==inicio){
            inicio=NULL;
        }
        else if (inicio->sig==fin && fin->ant==inicio){
            inicio=tmp->sig;
            inicio->sig=inicio;
            inicio->ant=inicio;
            fin=NULL;
        }
        else if(tmp==inicio){
            nodo *aux=new nodo();
            aux=inicio;
            aux->sig->ant=fin;
            aux->ant->sig=aux->sig;
            inicio=aux->sig;
            delete aux;
        }//esta al "inicio"
        else if(tmp==fin){
            nodo *aux=new nodo();

```

```

        aux=fin;
        aux->sig->ant=aux->ant;
        aux->ant->sig=inicio;
        fin=aux->ant;
        delete aux;
    }//esta al "final"
    else{
        nodo *aux=new nodo();
        aux=tmp;
        aux->sig->ant=aux->ant;
        aux->ant->sig=aux->sig;
        delete aux;
    }//es "intermedio"
    tmp=inicio;
} //fin elimina nodo

system("pause");
system("cls");
}while(op!=4 && inicio!=NULL);
if(inicio==NULL){cout<<"Al parecer ya no tienes mas
elementos\n";}
}

void mostrarLista(){
    int op=-1;
    tmp=inicio;
    do{
        cout<<"Dato Actual: "<<tmp->dato<<endl;
        cout<<"1.Mostrar dato siguiente\n";
        cout<<"2.Mostrar dato anterior\n";
    }

```

```
        cout<<"3.Salir\nSeleccion--->";
        cin>>op;
        if(op==1){tmp=tmp->sig;}
        if(op==2){tmp=tmp->ant;}

        system("pause");
        system("cls");
    }while(op!=3);
}
```

CODIGO LISTA CIRCULAR SIMPLE + PROBLEMA DE APLICACIÓN (LIBRO JOYANES) (NIVEL DE ENTENDIMIENTO INTERMEDIO)

II

//Estructuras Computacionales 2°A (PARCIAL 2)

/*

Ángel David Ortiz Quiroz	ID: 261481
--------------------------	------------

Ximena Rivera Delgadillo	ID: 261261
--------------------------	------------

Erick Ivan Ramirez Reyes	ID: 260806
--------------------------	------------

Diego Emanuel Saucedo Ortega	ID: 261230
------------------------------	------------

Jose Luis Sandoval Perez	ID: 261731
--------------------------	------------

Carlos Daniel Torres Macias	ID: 244543
-----------------------------	------------

*/

```
#include <iostream>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
using namespace std;
```

```
struct nodo{
```

```
    long int num;
```

```
    nodo *sig;
```

```
};
```

```

nodo *inicio=NULL;

void sumarLargos(int,int);
void insertarNum(int);

int main(){
    cout<<"PROBLEMA:\nPara representar un entero largo, de mas de
30 digitos, utilizar una lista circular\n";
    cout<<"cuyos nodos tienen como atributo dato un digito del
entero largo. Escribir un programa cuya entrada\n";
    cout<<"sea dos enteros largos y se obtenga su suma.\n";
    cout << "-----\n";
    -----\n";

    system("pause");
    system("cls");

    long int A,B;
    cout<<"Ingrese primer entero largo: "; cin>>A;
    cout<<"Ingrese segundo entero largo: "; cin>>B;
    system("pause");
    sumarLargos(A,B);
    cout<<"RESULTADO DE LA SUMA: ";
    nodo *t = new nodo();
    t=inicio;
    while(t!=NULL){
        cout<<t->num;
        t=t->sig;
    }//fin while
    cout<<endl;
    system("pause");
}

```



```

        system("cls");

        cout << "-----\n";
        cout << "PROGRAMA HECHO POR:" << endl;
        cout << "Angel David Ortiz Quiroz          ID : 261481" << endl;
        cout << "Ximena Rivera Delgadillo          ID : 261261" << endl;
        cout << "Erick Ivan Ramirez Reyes          ID : 260806" << endl;
        cout << "Diego Emanuel Saucedo Ortega        ID : 261230" << endl;
        cout << "Jose Luis Sandoval Perez          ID : 261731" << endl;
        cout << "Carlos Daniel Torres Macias        ID : 244543" << endl;
        system("pause");
return 0;
}

void sumarLargos(int A,int B){
    while(A>0 || B>0){
        int aux = (A%10) + (B%10);
        insertarNum(aux%10);
        A/=10; B/=10;
        if(aux>9){A+=aux/10;}
    }//fin while
}

void insertarNum(int N){
    if(inicio==NULL){
        inicio = new nodo();
        inicio->num = N;
        inicio->sig = NULL;
    }
}

```

```
else{
    nodo *tmp = new nodo();
    tmp->num = N;
    tmp->sig = inicio;
    inicio = tmp;
}
}
```

LISTAS CIRCULARES DOBLEMENTE ENLAZADAS

(NIVEL DE ENTENDIMIENTO AVANZADO)

```
//PROYECTO 2 parcial II
//PROGRAMA II
//Estructuras Computacionales 2ºA (PARCIAL 2)
/*
Ángel David Ortiz Quiroz          ID: 261481
Ximena Rivera Delgadillo          ID: 261261
Erick Ivan Ramirez Reyes          ID: 260806
Diego Emanuel Saucedo Ortega      ID: 261230
Jose Luis Sandoval Perez          ID: 261731
Carlos Daniel Torres Macias       ID: 244543
*/
```

```
#include <iostream>
using namespace std;
```

```
#define ASCENDENTE 1
#define DESCENDENTE 0
```

```
class nodo {
```

```

public:
    nodo(int v, nodo *sig = NULL, nodo *ant = NULL) :
        valor(v), siguiente(sig), anterior(ant) {}

private:
    int valor;
    nodo *siguiente;
    nodo *anterior;

friend class lista;
};

typedef nodo *pnodo;

class lista {
public:
    lista() : plista(NULL) {}
    ~lista();

    void Insertar(int v);
    void Borrar(int v);
    void Buscar(int v);
    bool ListaVacía() { return plista == NULL; }
    void Mostrar(int);
    void Siguiente();
    void Anterior();
    void Primero();
    void Ultimo();
    bool Actual() { return plista != NULL; }

```

```

    int ValorActual() { return plista->valor; }

private:
    pnode plista;
};

lista::~~lista() {
    pnode aux;

    Primero();
    while(plista) {
        aux = plista;
        plista = plista->siguiente;
        delete aux;
    }
}

void lista::Insertar(int v) {
    pnode nuevo;

    Primero();
    // Si la lista está vacía
    if(ListaVacia() || plista->valor > v) {
        // Asignamos a lista un nuevo nodo de valor v y
        // cuyo siguiente elemento es la lista actual
        nuevo = new nodo(v, plista);
        if(!plista) plista = nuevo;
        else plista->anterior = nuevo;
    }
}

```

```

else {
    // Buscar el nodo de valor menor a v
    // Avanzamos hasta el último elemento o hasta que el
siguiente tenga
    // un valor mayor que v
    while(plista->siguiente && plista->siguiente->valor <= v)
Siguiendo();
    // Creamos un nuevo nodo después del nodo actual
    nuevo = new nodo(v, plista->siguiente, plista);
    plista->siguiente = nuevo;
    if(nuevo->siguiente) nuevo->siguiente->anterior = nuevo;
}
}

```

```

void lista::Borrar(int v) {
    pnode nodo;

    nodo = plista;
    while(nodo && nodo->valor < v) nodo = nodo->siguiente;
    while(nodo && nodo->valor > v) nodo = nodo->anterior;

    if(!nodo || nodo->valor != v) return;
    // Borrar el nodo

    if(nodo->anterior) // no es el primer elemento
        nodo->anterior->siguiente = nodo->siguiente;
    if(nodo->siguiente) // no es el último nodo
        nodo->siguiente->anterior = nodo->anterior;
    delete nodo;
}

```

```

void lista::Mostrar(int orden) {
    pnode nodo;
    if(orden == ASCENDENTE) {
        Primero();
        nodo = plista;
        while(nodo) {
            cout << nodo->valor << "-> ";
            nodo = nodo->siguiente;
        }
    }
    else {
        Ultimo();
        nodo = plista;
        while(nodo) {
            cout << nodo->valor << "-> ";
            nodo = nodo->anterior;
        }
    }
    cout << endl;
}

void lista::Siguiete() {
    if(plista) plista = plista->siguiente;
}

void lista::Anterior() {
    if(plista) plista = plista->anterior;
}

```

```
}
```

```
void lista::Primero() {  
    while(plista && plista->anterior) plista = plista->anterior;  
}
```

```
void lista::Ultimo() {  
    while(plista && plista->siguiente) plista = plista->siguiente;  
}
```

```
int main() {  
    int op, dato;  
    lista Lista;  
  
    do {  
        cout << "\t.:MENU:.\n";  
        cout << "1. Insertar elementos a la lista de id's\n";  
        cout << "2. Mostrar elementos de lista de id's en orden  
ascendente\n";  
        cout << "3. Mostrar elementos de lista de id's en orden  
descendente\n";  
        cout << "4. Eliminar un elemento de la lista\n";  
        cout << "5. Salir\n";  
        cin >> op;  
  
        switch (op) {  
            case 1: cout << "Digite un numero: ";  
                    cin >> dato;  
                    Lista.Insertar(dato);  
                    system("pause");
```



```

        break;
    case 2: Lista.Mostrar(ASCENDENTE);
        cout << "\n";
        system("pause");
        break;
    case 3: Lista.Mostrar(DESCENDENTE);
        cout << "\n";
        system("pause");
        break;
    case 4: cout << "Digite el ID que desea eliminar: ";
        cin >> dato;
        Lista.Borrar(dato);
        cout << "\nListo";
        system("pause");
    }
    system("cls");
} while (op != 5);

```

```

        cout << "-----\n";
        cout << "PROGRAMA HECHO POR:" << endl;
        cout << "Angel David Ortiz Quiroz          ID : 261481" << endl;
        cout << "Ximena Rivera Delgadillo          ID : 261261" << endl;
        cout << "Erick Ivan Ramirez Reyes          ID : 260806" << endl;
        cout << "Diego Emanuel Saucedo Ortega      ID : 261230" << endl;
        cout << "Jose Luis Sandoval Perez          ID : 261731" << endl;
        cout << "Carlos Daniel Torres Macias       ID : 244543" << endl;

```

```
Lista.Primer();  
cout << "Primer: " << Lista.ValorActual() << endl;  
  
Lista.Ultimo();  
cout << "Ultimo: " << Lista.ValorActual() << endl;  
  
return 0;  
}
```