|  |  |  |
| --- | --- | --- |
| **CARRERA:**  Ingeniería de Software | **GUÍA**  No. 02 | **TIEMPO ESTIMADO:**  1 mes |
| **ASIGNATURA:**  Estructura de datos  NRC: 2967 | **FECHA DE ELABORACION:** 26-01-2020  **SEMESTRE**: septiembre 2019 – febrero 2020 | |
| **TÍTULO:**  Listas | **DOCENTE:** Ing. Fernando Solis | |

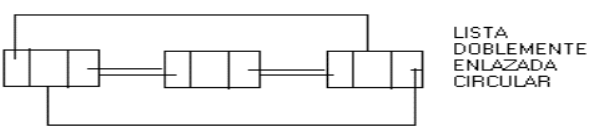
**OBJETIVO**

Comprender el uso de listas (listas simples, listas dobles, listas doblemente circulares, etc.) para crear un pequeño juego con listas doblemente circulares.

**INSTRUCCIONES**

**Listas Doblemente Circulares**

En una lista enlazada doblemente circular, cada nodo tiene dos enlaces, similares a los de la lista doblemente enlazada, excepto que el enlace anterior del primer nodo apunta al último y el enlace siguiente del último nodo, apunta al primero. Como en una lista doblemente enlazada, las inserciones y eliminaciones pueden ser hechas desde cualquier punto con acceso a algún nodo cercano. Aunque estructuralmente una lista circular doblemente enlazada no tiene ni principio ni fin, un puntero de acceso externo puede establecer el nodo apuntado que está en la cabeza o al nodo cola, y así mantener el orden tan bien como en una lista doblemente enlazada.



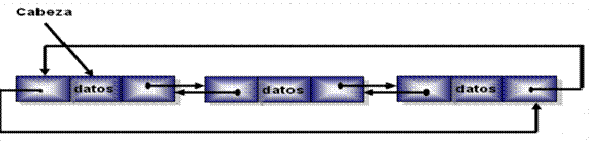
Una lista doble circular es una estructura donde el último elemento tiene como referencia siguiente al primer elemento y la referencia al anterior del primer elemento de la lista también es el último. Cada elemento esta doblemente enlazado.

A través del uso de listas dobles podemos acceder a los datos recorriendo los hacia delante hasta el final o hacia atrás hasta el inicio.

**Operaciones básicas**

Las operaciones básicas de una lista circular doble son:

* Insertar un nodo con dato X en la lista realizando una inserción al principio o al final de la lista.
* Eliminar un nodo de la lista, puede ser según la posición o por el dato
* Buscar un elemento en la lista
* Obtener la posición del nodo en la lista
* Imprimir los elementos de la lista



Ejemplo de lista enlazada doble circular

Un ejemplo donde se almacenan números y vemos como la estructura donde el último

**ACTIVIDADES**

1. **Ubicación de recursos**
2. El grupo está conformado por Maria Belen Ceron y Alex Chicaiza
3. Se utilizo CodeBlocks
4. **Planteamiento del problema**

Realizar el juego de Tetris utilizando listas doblemente circulares donde, si caen dos números iguales estos se unirán y se suprimirán, de lo contrario se seguirán acumulando.

1. **Entregable (s)**

**Clase Nodo.h**

#if !defined(\_\_Tetris\_Nodo\_h)

#define \_\_Tetris\_Nodo\_h

class Nodo

**{**

public**:**

Nodo**();**

**~**Nodo**();**

void setNumero**(**int**);**

int getNumero**();**

void setAnteriorDireccion**(**Nodo**\*);**

Nodo**\*** getAnteriorDireccion**();**

void setSiguienteDireccion**(**Nodo**\*);**

Nodo**\*** getSiguienteDireccion**();**

private**:**

int numero**;**

Nodo**\*** anteriorDireccion**;**

Nodo**\*** siguienteDireccion**;**

**};**

#endif

**Clase Nodo.cpp**

#include "Nodo.h"

////////////////////////////////////////////////////////////////////////

// Name: Nodo::Nodo()

// Purpose: Implementation of Nodo::Nodo()

// Return:

////////////////////////////////////////////////////////////////////////

Nodo**::**Nodo**()**

**{**

**}**

////////////////////////////////////////////////////////////////////////

// Name: Nodo::~Nodo()

// Purpose: Implementation of Nodo::~Nodo()

// Return:

////////////////////////////////////////////////////////////////////////

Nodo**::~**Nodo**()**

**{**

// TODO : implement

**}**

////////////////////////////////////////////////////////////////////////

// Name: Nodo::getNumero()

// Purpose: Implementation of Nodo::getNumero()

// Return: int

////////////////////////////////////////////////////////////////////////

int Nodo**::**getNumero**(**void**)**

**{**

**return** numero**;**

**}**

////////////////////////////////////////////////////////////////////////

// Name: Nodo::setNumero(int newNumero)

// Purpose: Implementation of Nodo::setNumero()

// Parameters:

// - newNumero

// Return: void

////////////////////////////////////////////////////////////////////////

void Nodo**::**setNumero**(**int newNumero**)**

**{**

numero **=** newNumero**;**

**}**

////////////////////////////////////////////////////////////////////////

// Name: Nodo::getAnteriorDireccion()

// Purpose: Implementation of Nodo::getAnteriorDireccion()

// Return: Nodo\*

////////////////////////////////////////////////////////////////////////

Nodo**\*** Nodo**::**getAnteriorDireccion**(**void**)**

**{**

**return** anteriorDireccion**;**

**}**

////////////////////////////////////////////////////////////////////////

// Name: Nodo::setAnteriorDireccion(Nodo\* newAnteriorDireccion)

// Purpose: Implementation of Nodo::setAnteriorDireccion()

// Parameters:

// - newAnteriorDireccion

// Return: void

////////////////////////////////////////////////////////////////////////

void Nodo**::**setAnteriorDireccion**(**Nodo**\*** newAnteriorDireccion**)**

**{**

anteriorDireccion **=** newAnteriorDireccion**;**

**}**

////////////////////////////////////////////////////////////////////////

// Name: Nodo::getSiguienteDireccion()

// Purpose: Implementation of Nodo::getSiguienteDireccion()

// Return: Nodo\*

////////////////////////////////////////////////////////////////////////

Nodo**\*** Nodo**::**getSiguienteDireccion**(**void**)**

**{**

**return** siguienteDireccion**;**

**}**

////////////////////////////////////////////////////////////////////////

// Name: Nodo::setSiguienteDireccion(Nodo\* newSiguienteDireccion)

// Purpose: Implementation of Nodo::setSiguienteDireccion()

// Parameters:

// - newSiguienteDireccion

// Return: void

////////////////////////////////////////////////////////////////////////

void Nodo**::**setSiguienteDireccion**(**Nodo**\*** newSiguienteDireccion**)**

**{**

siguienteDireccion **=** newSiguienteDireccion**;**

**}**

**Clase Cifrado.h**

#pragma once

#include <iostream>

#include <stdio.h>

#include <windows.h>

#include <conio.h>

#include<string>

**using** **namespace** std**;**

class Cifrado

**{**

public**:**

void encriptado**();**

void desencriptado**();**

private**:**

FILE**\*** archivo**,** **\*** nuevo**;**

**};**

**Clase Cifrado.cpp**

#include "Cifrado.h"

void Cifrado**::**encriptado**()** **{**

char car**;**

archivo **=** fopen**(**"solucion.txt"**,** "rb"**);**

nuevo **=** fopen**(**"SolucionActualizada.txt"**,** "wb"**);**

**if** **(**nuevo **==** **NULL)** **{**

cout **<<** "Error al abrir el archivo nuevo\n"**;**

**}**

**if** **(**archivo **==** **NULL)** **{**

cout **<<** "Error al abrir el archivo\n"**;**

**}**

**while** **(**fread**(&**car**,** 1**,** 1**,** archivo**)** **!=** 0**)** **{**

car **+=** 20**;**

fwrite**(&**car**,** 1**,** 1**,** nuevo**);**

**}**

fclose**(**nuevo**);**

fclose**(**archivo**);**

**}**

void Cifrado**::**desencriptado**()** **{**

char car**;**

archivo **=** fopen**(**"solucion.txt"**,** "wb"**);** //FALTA

nuevo **=** fopen**(**"SolucionActualizada.txt"**,** "rb"**);**

**if** **(**nuevo **==** **NULL)** **{**

cout **<<** "Error al abrir el archivo nuevo\n"**;**

**}**

**if** **(**archivo **==** **NULL)** **{**

cout **<<** "Error al abrir el archivo\n"**;**

**}**

**while** **(**fread**(&**car**,** 1**,** 1**,** nuevo**)** **!=** 0**)** **{**

car **-=** 20**;**

fwrite**(&**car**,** 1**,** 1**,** archivo**);**

**}**

fclose**(**nuevo**);**

fclose**(**archivo**);**

**}**

**Clase ListaDoble.h**

#include "Nodo.h"

#include <iostream>

#define \_\_Tetris\_Nodo\_h

class ListaDoble

**{**

public**:**

ListaDoble**()**

**{**

lista**=NULL;**

contadorNodo**=**0**;**

puntaje**=**0**;**

**}**

void insertarAlFinal**(**int**);**

void impresion**();**

void impresionLista**();**

void generarLista**();**

void listaTetris**(**int**,**int**);**

void imprimir**();**

void juegoTetris**();**

int posicionAIndice**(**int x**);**

void deleteNumber**(**int**,**int **);**

void siguienteNumero**(**int**,**Nodo **\*** **);**

void deleteNodo**(**Nodo **\***Actual**);**

private**:**

Nodo **\***lista**;**

std**::**string jugador**;**

int contadorNodo**;**

int puntaje**;**

**};**

void ListaDoble**::**impresion**()**

**{**

Nodo**\*** Aux**=new** Nodo**();**

Aux**=**lista**;**

**if(**Aux **==** **NULL){**

**}**

**else**

**{**

gotoxy**(**20**,**15**);**

cout**<<**"Jugador: "**<<**jugador**;**

gotoxy**(**20**,**17**);**

cout**<<**"lista"**;**

gotoxy**(**20**,**19**);**

**while(**Aux **!=** **NULL)**

**{**

printf**(**"%d, "**,**Aux**->**getNumero**());**

Aux**=**Aux**->**getSiguienteDireccion**();**

**}**

gotoxy**(**20**,**21**);**printf**(**"Su Puntaje es: %d \n\n"**,**puntaje**);**

**}**

**}**

void ListaDoble**::**insertarAlFinal**(**int numero**)**

**{**

Nodo**\*** Nuevo**=new** Nodo**();**

Nodo**\*** Actual**=new** Nodo**();**

**if(**contadorNodo**==**0**)**

**{**

Actual**->**setNumero**(**numero**);**

Actual**->**setAnteriorDireccion**(NULL);**

Actual**->**setSiguienteDireccion**(NULL);**

lista**=**Actual**;**

**}**

**else**

**{**

Actual**=**lista**;**

**while(**Actual**->**getSiguienteDireccion**()!=NULL)**

**{**

Actual**=**Actual**->**getSiguienteDireccion**();**

**}**

Nuevo**->** setNumero**(**numero**);**

Nuevo**->** setSiguienteDireccion**(NULL);**

Actual**->**setSiguienteDireccion**(**Nuevo**);**

Nuevo**->** setAnteriorDireccion**(**Actual**);**

**}**

contadorNodo**++;**

**}**

void ListaDoble**::**generarLista**()**

**{**

Nodo**\*** Aux**=new** Nodo**();**

int numero**;**

**for(**int i**=**0**;**i**<**3**;**i**++)**

**{**

numero**=**rand**()%**10**;**

insertarAlFinal**(**numero**);**

**}**

**}**

void ListaDoble**::**listaTetris**(**int posicion**,**int numero**)**

**{**

Nodo **\***Aux**=new** Nodo**();**

Nodo **\***Actual**=new** Nodo**();**

Nodo **\***Nuevo**=**lista**;**

int cont**;**

**while(**Nuevo**!=NULL)**

**{**

**if(**numero**==**Nuevo**->**getNumero**())**

**{**

cont**++;**

**break;**

**}**

Nuevo**=**Nuevo**->**getSiguienteDireccion**();**

**}**

**switch(**posicion**)**

**{**

**case** 1**:**

**if(**cont**>**0**)**

**{**

Nuevo**=**Nuevo**->**getSiguienteDireccion**();**

Nuevo**->**setAnteriorDireccion**(NULL);**

lista**=**Nuevo**;**

contadorNodo**--;**

**}**

**else**

**{**

Aux**->**setNumero**(**numero**);**

Aux**->**setSiguienteDireccion**(**lista**);**

Aux**->**setAnteriorDireccion**(NULL);**

Actual**->**setAnteriorDireccion**(**Aux**);**

lista**=**Aux**;**

contadorNodo**++;**

**}**

**break;**

**}**

**}**

int ListaDoble**::**posicionAIndice**(**int x**){**

**return** **(**x**-**31**)/**4**;**

**}**

void ListaDoble**::**deleteNodo**(**Nodo **\***Actual**){**

Nodo**\*** Siguiente**=new** Nodo**();**

Nodo**\*** Anterior**=new** Nodo**();**

**if(**Actual**->**getAnteriorDireccion**()!=NULL&&**Actual**->**getSiguienteDireccion**()!=NULL){**

Anterior**=**Actual**->**getAnteriorDireccion**();**

Siguiente**=**Actual**->**getSiguienteDireccion**();**

Anterior**->**setSiguienteDireccion**(**Siguiente**);**

Siguiente**->**setAnteriorDireccion**(**Anterior**);**

**delete(**Actual**);**

**}else** **if(**Actual**->**getAnteriorDireccion**()==NULL){**

lista**=**Actual**->**getSiguienteDireccion**();**

**delete(**Actual**);**

**}else** **if(**Actual**->**getSiguienteDireccion**()==NULL){**

Anterior**=**Actual**->**getAnteriorDireccion**();**

Anterior**->**setSiguienteDireccion**(NULL);**

**delete(**Actual**);**

**}**

**}**

void ListaDoble**::**siguienteNumero**(**int numero**,**Nodo **\***aux**){**

**if(**aux**==NULL||(**aux**->**getNumero**()!=(**numero**+**1**))){**

**return** **;**

**}else{**

**if(**aux**->**getNumero**()==(**numero**+**1**)){**

puntaje**++;**

deleteNodo**(**aux**);**

**}**

**}**

**}**

void ListaDoble**::**deleteNumber**(**int indice**,**int numero**){**

int cont**=**0**;**

Nodo**\*** Actual**=new** Nodo**();**

Actual**=**lista**;**

**while(**Actual**!=NULL){**

**if(**indice**==**cont**){**

**if(**Actual**->**getNumero**()==**numero**)**//borra el numero

**{**

contadorNodo**--;**

puntaje**++;**

deleteNodo**(**Actual**);**

siguienteNumero**(**numero**,**Actual**);**

**}**

**else**

**{**

cout**<<**numero**<<**endl**;**

insertarAlFinal**(**numero**);**

**}**

Actual**=NULL;**

**}**

**else**

**{**

Actual**=**Actual**->**getSiguienteDireccion**();**

**}**

cont**++;**

**}**

**}**

void ListaDoble**::**juegoTetris**()**

**{**

bool primera**=true;**

char tecla**=**0**;**

int x**=**31**,**y**=**4**;**

int col**,**fil**,**j**=**0**;**

int numero**;**

char nombre**[**200**];**

int i**=**0**;**

srand**(**time**(NULL));**

lista**=NULL;**

contadorNodo**=**puntaje**=**0**;**

gotoxy**(**20**,**15**);**

printf**(**"NOMBRE DEL JUGADOR: "**);**

cin**>>**jugador**;**

fflush**(**stdin**);**

system**(**"cls"**);**

generarLista**();**

margen**();**

impresionLista**();**

gotoxy**(**110**,**20**);**

cout**<<**"Jugador: "**<<**jugador**;**

gotoxy**(**110**,**21**);**cout**<<**"lista:"**;**

imprimir**();**

**do**

**{**

Sleep**(**200**);**

borrar**(**38**);**

**if(**primera**)**

**{**

numero**=**0**+**rand**()%**9**;**

primera**=false;**

gotoxy**(**4**,**45**);**

**}**

**if((**x**==**10**)&&(**y**==**23**))** /// aqui mandar cuando encuentre el numero

**{**

listaTetris**(**1**,**numero**);**

numero**=**0**+**rand**()%**9**;**

x**=**34**;**y**=**3**;**

borrar**(**39**);**

impresionLista**();**

imprimir**();**

**}**

//impresionLista();

gotoxy**(**x**,**y**);**

numbers**(**numero**,**x**,**y**);**

color**(**15**);**

y**++;**

**if(**y**>=**37**){**

deleteNumber**(**posicionAIndice**(**x**),**numero**);**

y**=**4**;**

primera**=true;**

borrar**(**39**);**

impresionLista**();**

imprimir**();**

**}**

**if(**kbhit**()){**

**switch(**getch**())**

**{**

**case** TECLA\_DERECHA**:**

x**+=**4**;**

**if(**x**>**95**)**

x**=**95**;**

**break;**

**case** TECLA\_IZQUIERDA**:**

x**-=**4**;**

**if(**x**<**31**)**

x**=**31**;**

**break;**

**case** TECLA\_ENTER**:**

tecla**=**TECLA\_ENTER**;**

**break;**

**}**

**}**

**if(**contadorNodo**==**18**)**

tecla**=**TECLA\_ENTER**;**

**}while(**tecla**!=**TECLA\_ENTER**);**

system**(**"cls"**);**

**}**

void ListaDoble**::**impresionLista**()**

**{**

int x**=**31**,**y**;**

Nodo**\*** Aux**=new** Nodo**();**

Aux**=**lista**;**

**while(**Aux **!=** **NULL)**

**{**

y**=**38**;**

**switch(**Aux**->**getNumero**())**

**{**

**case** 0**:**

color**(**1**);**// num 0 azul

gotoxy**(**x**-**1**,**y**-**1**);**

printf**(**"| |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| 0 |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| |"**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 1**:**

color**(**2**);**

gotoxy**(**x**-**1**,**y**-**1**);**

printf**(**"| |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| 1 |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| |"**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 2**:**

color**(**11**);**

gotoxy**(**x**-**1**,**y**-**1**);**

printf**(**"| |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| 2 |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| |"**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 3**:**

color**(**12**);**

gotoxy**(**x**-**1**,**y**-**1**);**

printf**(**"| |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| 3 |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| |"**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 4**:**

color**(**5**);**

gotoxy**(**x**-**1**,**y**-**1**);**

printf**(**"| |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| 4 |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| |"**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 5**:**

color**(**6**);**

gotoxy**(**x**-**1**,**y**-**1**);**

printf**(**"| |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| 5 |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| |"**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 6**:**

color**(**15**);**

gotoxy**(**x**-**1**,**y**);**

printf**(**"| |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| 6 |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| |"**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 7**:**

color**(**8**);**

gotoxy**(**x**-**1**,**y**-**1**);**

printf**(**"| |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| 7 |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| |"**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 8**:**

color**(**13**);**

gotoxy**(**x**-**1**,**y**-**1**);**

printf**(**"| |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| 8 |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| |"**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 9**:**

color**(**10**);**

gotoxy**(**x**-**1**,**y**-**1**);**

printf**(**"| |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| 9 |"**);**gotoxy**(**x**,**y**++);**

printf**(**"| |"**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**}**

x**+=**4**;**//separacion entre cubos

Aux**=**Aux**->**getSiguienteDireccion**();**

**}**

**}**

void ListaDoble**::**imprimir**()**

**{**

Nodo**\*** Aux**=new** Nodo**();**

Aux**=**lista**;**

gotoxy**(**110**,**22**);**

cout**<<**" "**;**

gotoxy**(**110**,**22**);**

fstream enter**;**

enter**.**open**(**"solucion.txt"**,**fstream**::**app**);**

enter**<<**"solucion "**<<**"\*"**<<**endl**<<**endl**;**

**while(**Aux **!=** **NULL)**

**{**

cout**<<**Aux**->**getNumero**()<<**", "**;**

enter**<<**Aux**->**getNumero**();**

Aux**=**Aux**->**getSiguienteDireccion**();**

**}**

enter**<<**endl**;**

**}**

**Clase PersonalLibrary.h**

#include <stdio.h>

#include <iostream>

#include <stdlib.h>

#include <conio.h>

#include <windows.h>

#include <ctype.h>

#include <string.h>

#include <time.h>

#include <pthread.h>

#include <fstream>

**using** **namespace** std**;**

#define TECLA\_ARRIBA 72

#define TECLA\_ABAJO 80

#define TECLA\_DERECHA 77

#define TECLA\_IZQUIERDA 75

#define TECLA\_ENTER 13

void gotoxy**(**int x**,** int y**)**

**{**

HANDLE hCon**;**

hCon**=**GetStdHandle**(**STD\_OUTPUT\_HANDLE**);**

COORD dwPos**;**

dwPos**.**X**=**x**;**

dwPos**.**Y**=**y**;**

SetConsoleCursorPosition**(**hCon**,**dwPos**);**

**}**

void color**(**int x**)**

**{**

SetConsoleTextAttribute**(**GetStdHandle **(**STD\_OUTPUT\_HANDLE**),**x**);**

**}**

void AltEnter**()**

**{**

keybd\_event**(**VK\_MENU**,**

0x38**,**

0**,**

0**);**

keybd\_event**(**VK\_RETURN**,**

0x1c**,**

0**,**

0**);**

keybd\_event**(**VK\_RETURN**,**

0x1c**,**

KEYEVENTF\_KEYUP**,**

0**);**

keybd\_event**(**VK\_MENU**,**

0x38**,**

KEYEVENTF\_KEYUP**,**

0**);**

**}**

void numbers**(**int numero**,**int x**,**int y**)**

**{**

**switch(**numero**){**

**case** 0**:**

color**(**1**);**// num 0 azul

gotoxy**(**x**,**y**-**1**);**

printf**(**" ====="**);**gotoxy**(**x**,**y**++);**

printf**(**"|| 0 ||"**);**gotoxy**(**x**,**y**++);**

printf**(**" ====="**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 1**:**

color**(**2**);**

gotoxy**(**x**,**y**-**1**);**

printf**(**" ====="**);**gotoxy**(**x**,**y**++);**

printf**(**"|| 1 ||"**);**gotoxy**(**x**,**y**++);**

printf**(**" ====="**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 2**:**

color**(**11**);**

gotoxy**(**x**,**y**-**1**);**

printf**(**" ====="**);**gotoxy**(**x**,**y**++);**

printf**(**"|| 2 ||"**);**gotoxy**(**x**,**y**++);**

printf**(**" ====="**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 3**:**

color**(**12**);**

gotoxy**(**x **,**y**-**1**);**

printf**(**" ====="**);**gotoxy**(**x**,**y**++);**

printf**(**"|| 3 ||"**);**gotoxy**(**x**,**y**++);**

printf**(**" ====="**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 4**:**

color**(**5**);**

gotoxy**(**x**,**y**-**1**);**

printf**(**" ====="**);**gotoxy**(**x**,**y**++);**

printf**(**"|| 4 ||"**);**gotoxy**(**x**,**y**++);**

printf**(**" ====="**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 5**:**

color**(**6**);**

gotoxy**(**x**,**y**-**1**);**

printf**(**" ====="**);**gotoxy**(**x**,**y**++);**

printf**(**"|| 5 ||"**);**gotoxy**(**x**,**y**++);**

printf**(**" ====="**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 6**:**

color**(**15**);**

gotoxy**(**x**,**y**-**1**);**

printf**(**" ====="**);**gotoxy**(**x**,**y**++);**

printf**(**"|| 6 ||"**);**gotoxy**(**x**,**y**++);**

printf**(**" ====="**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 7**:**

color**(**8**);**

gotoxy**(**x**,**y**-**1**);**

printf**(**" ====="**);**gotoxy**(**x**,**y**++);**

printf**(**"|| 7 ||"**);**gotoxy**(**x**,**y**++);**

printf**(**" ====="**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 8**:**

gotoxy**(**x**,**y**-**1**);**

printf**(**" ====="**);**gotoxy**(**x**,**y**++);**

printf**(**"|| 8 ||"**);**gotoxy**(**x**,**y**++);**

printf**(**" ====="**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**case** 9**:**

color**(**10**);**

gotoxy**(**x**,**y**-**1**);**

printf**(**" ====="**);**gotoxy**(**x**,**y**++);**

printf**(**"|| 9 ||"**);**gotoxy**(**x**,**y**++);**

printf**(**" ====="**,**219**,**219**,**219**,**219**);**gotoxy**(**x**,**y**++);**

**break;**

**}**

**}**

void margen**()**

**{**

color**(**15**);**

**for(**int i**=**30**;** i **<** 99**;** i**++)**

**{**

//PARTE SUPERIOR

gotoxy **(**i**,** 3**);** printf**(**"="**);**

//PARTE INFERIOR

gotoxy**(**i**,** 40**);** printf**(**"="**);**

**}**

**for(**int j**=**3**;** j **<** 40**;** j**++)**

**{**

//PARTE IZQUIERDA

gotoxy **(**29**,**j**);** printf**(**"||"**);**

//PARTE DERECHA

gotoxy**(**99**,**j**);** printf**(**"||"**);**

**}**

//ESQUINAS

color**(**15**);**

gotoxy**(**99**,**40**);** printf**(**"||"**);**

**}**

void borrar**(**int numero**)**

**{**

color**(**15**);**

**for(**int i**=**31**;** i **<** 99**;** i**++)**

**{**

//PARTE SUPERIOR

**for(**int j**=**4**;**j**<**numero**;**j**++){**

gotoxy **(**i**,** j**);** printf**(**" "**);**

**}**

**}**

**}**

**Clase Main**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Universidad de las Fuerzas Armadas ESPE

\* Maria Belen Ceron, Alex Chicaiza

\* NRC: 2967

\* Fecha creacion: 08/11/2019

\* Fecha ultima modificacion: 12/12/2019

\* Docente: Ing. Fernando Solis

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <windows.h>

#include <conio.h>

#include "Cifrado.h"

#include "PersonalLibrary.h"

#include "ListaDoble.h"

#define ARRIBA 72

#define IZQUIERDA 75

#define DERECHA 77

#define ABAJO 80

int cont **=** 15**;**

**using** **namespace** std**;**

int i**=**21**;**

Cifrado si**;**

ListaDoble ObjJuego**;**

void seleccionarOpcion**(**char tecla**);**

void menuinicio**(**char tecla**);**

void dibujarCuadro**()** **{**

cout **<<** "\t\t==========================================================\n"**;**

cout **<<** "\t\t|| ||\n"**;**

cout **<<** "\t\t|| ||\n"**;**

cout **<<** "\t\t|| ||\n"**;**

cout **<<** "\t\t|| ||\n"**;**

cout **<<** "\t\t|| ||\n"**;**

cout **<<** "\t\t|| ||\n"**;**

cout **<<** "\t\t|| ||\n"**;**

cout **<<** "\t\t|| ||\n"**;**

cout **<<** "\t\t==========================================================\n"**;**

**}**

void bran**(**int cont**)** **{**

gotoxy**(**18**,** cont**);**

SetConsoleTextAttribute**(**GetStdHandle**(**STD\_OUTPUT\_HANDLE**),** 3**);**

cout **<<** ">"**;**

**}**

void gotoxy**(**short posicionx**,** short posiciony**)** **{**

COORD coordenadaPosicion **=** **{** posicionx**,** posiciony **};**

SetConsoleCursorPosition**(**GetStdHandle**(**STD\_OUTPUT\_HANDLE**),** coordenadaPosicion**);**

**}**

void empezarProgramas**(**string nombre**)** **{**

ShellExecute**(NULL,** "open"**,** nombre**.**c\_str**(),** **NULL,** **NULL,** SW\_SHOWNORMAL**);**

**}**

void menu**()** **{**

//system("Color 7A");

gotoxy**(**0**,** 0**);**

cout **<<** "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**;**

gotoxy**(**0**,** 2**);**

cout **<<** "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**;**

gotoxy**(**0**,** 3**);** dibujarCuadro**();**

gotoxy**(**0**,** 13**);** dibujarCuadro**();**

gotoxy**(**10**,** 5**);** cout **<<** "\t\tUNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE\n\t\t\t\* Alex Chicaiza\n\t\t\t\* Maria Belen Ceron\n\t\t\tNRC: 2967\n\t\t\tDocente: Ing. Fernando Solis" **<<** endl**;**

gotoxy**(**42**,** 14**);** cout **<<** "MENU" **<<** endl**;**

gotoxy**(**18**,** 15**);** cout **<<** " 1- JUGAR\n"**;**

gotoxy**(**18**,** 16**);** cout **<<** " 2- DESARROLLADORES\n"**;**

gotoxy**(**18**,** 17**);** cout **<<** " 3- GENERAR PDF\n"**;**

gotoxy**(**18**,** 18**);** cout **<<** " 4- CIFRADO\n"**;**

gotoxy**(**18**,** 19**);** cout **<<** " 5- MOSTRAR IMAGEN EN CONSOLA\n"**;**

gotoxy**(**18**,** 20**);** cout **<<** " 6- AYUDA\n"**;**

gotoxy**(**18**,** 21**);** cout **<<** " 7- SALIR\n"**;**

gotoxy**(**18**,** cont**);**

//bran(cont);

cout **<<** ">"**;**

**if** **(**\_kbhit**())** **{**

**switch** **(**\_getch**())** **{**

**case** ARRIBA**:**

**if** **(**cont **<** 16**)** **{**

cont **=** 22**;**

**}**

cont**--;**

**break;**

**case** ABAJO**:**

**if** **(**cont **>** 20**)** **{**

cont **=** 14**;**

**}**

cont**++;**

**break;**

**case** 13**:**

**switch** **(**cont**)** **{**

**case** 15**:**

system**(**"cls"**);**

ObjJuego**.**juegoTetris**();**

system**(**"pause"**);**

**break;**

**case** 16**:**

system**(**"cls"**);**

empezarProgramas**(**"Desarrolladores.png"**);**

system**(**"pause"**);**

**break;**

**case** 17**:**

system**(**"cls"**);**

system**(**"txt2pdf.exe solucion.txt respuestas.pdf -oao -pfs60 -pps43 -ptc0 -width3000 -height2000"**);**

system**(**"pause"**);**

**break;**

**case** 18**:**

system**(**"cls"**);**

cout**<<**"Encriptador\n"**;**

si**.**encriptado**();**

cout**<<**"DEncriptador\n"**;**

si**.**desencriptado**();**

Sleep**(**200**);**

//system("pause");

**break;**

**case** 19**:**

system**(**"cls"**);**

system**(**"pause"**);**

**break;**

**case** 20**:**

system**(**"cls"**);**

empezarProgramas**(**"Tetris.chm"**);**

system**(**"pause"**);**

**break;**

**case** 21**:**

system**(**"cls"**);**

std**::**cout **<<** "\tGracias por usar nuestra aplicacion" **<<** std**::**endl**;**

exit**(**0**);**

**default:**

**break;**

**}**

system**(**"cls"**);**

menu**();**

//gotoxy(3, 1);

**break;**

**}**

**}**

**}**

//ESTA PARTE

void subMenu**()** **{**

int cont2 **=** 4**;**

gotoxy**(**0**,**4**);** cout **<<** "1. Encriptar" **<<** endl**;**

gotoxy**(**0**,**5**);** cout **<<** "2. DesEncriptar" **<<** endl**;**

gotoxy**(**0**,** cont2**);**

cout **<<** ">"**;**

**if** **(**\_kbhit**())** **{**

**switch** **(**\_getch**())** **{**

**case** ARRIBA**:**

**if** **(**cont2 **<** 4**)** **{**

cont2 **=** 5**;**

**}**

cont2**--;**

**break;**

**case** ABAJO**:**

**if** **(**cont2 **>** 5**)** **{**

cont2 **=** 4**;**

**}**

cont2**++;**

**break;**

**case** 13**:**

**switch** **(**cont2**)** **{**

**case** 4**:**

system**(**"cls"**);**

cout **<<** "hola"**;**

system**(**"pause"**);**

**break;**

**case** 5**:**

system**(**"cls"**);**

cout **<<** "Adios"**;**

system**(**"pause"**);**

**default:**

**break;**

**}**

system**(**"cls"**);**

subMenu**();**

**break;**

**}**

**}**

system**(**"pause"**);**

**}**

int main**()**

**{**

AltEnter**();**

char tecla**;**

int a**,** b**,** c**,** n **=** 0**,** letra **=** 39**,** pos **=** 1**,** cont **=** 39**,** aux**,** cont1 **=** 39**,** auxg**;**

char t**[**50**]** **=** "BIENVENIDO AL JUEGO DEL TETRIS"**,** auxt**[**39**]** **=** " "**;**

gotoxy**(**0**,** 0**);**

cout **<<** "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**;**

gotoxy**(**0**,** 2**);**

cout **<<** "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**;**

gotoxy**(**0**,** 3**);** dibujarCuadro**();**

gotoxy**(**0**,** 13**);** dibujarCuadro**();**

**do** **{**

**for** **(**a **=** 0**;** a **<** 30**;** a**++)** **{**

aux **=** pos**;**

**for** **(**b **=** 39**;** b **>** cont**;** b**--)** **{**

gotoxy**(**pos**,** 1**);**

cout **<<** t**[**b**];**

menu**();**

pos**--;**

**}**

aux**++;**

pos **=** aux**;**

cont**--;**

Sleep**(**75**);**

**if** **(**a **==** 40**)** **{**

**break;**

**}**

**}**

**for** **(**a **=** 3**;** a **<** 70**;** a**++)** **{**

gotoxy**(**a **-** 1**,** 1**);**

cout **<<** " "**;**

gotoxy**(**a**,** 1**);**

cout **<<** t**;**

menu**();**

Sleep**(**75**);**

**}**

pos **=** 70**;**

auxg **=** 69**;**

**for** **(**a **=** 0**;** a **<** 41**;** a**++)** **{**

gotoxy**(**auxg**,** 1**);**

cout **<<** " "**;**

aux **=** pos**;**

**for** **(**b **=** 0**;** b **<=** cont1**;** b**++)** **{**

gotoxy**(**pos**,** 1**);**

cout **<<** t**[**b**];**

pos**++;**

menu**();**

**}**

cont1**--;**

aux**++;**

pos **=** aux**;**

auxg**++;**

Sleep**(**75**);**

**}**

cont1 **=** 39**;**

letra **=** 39**;**

pos **=** 1**;**

cont **=** 39**;**

c **=** 0**;**

**}** **while** **(**c **=** 1**);**

system**(**"pause>null"**);**

**return** 0**;**

**}**

void seleccionarOpcion**(**char tecla**){**

gotoxy**(**30**,** i**);** cout**<<**" "**;**

**if(** tecla **==** ABAJO **&&** i **<=**24 **){**

**if(**i**==**24**){**

i**=**20**;**

**}**

i**++;**

**}**

**if(** tecla **==** ARRIBA **&&** i**>=** 21**){**

**if(**i**==**21**){**

i**=**25**;**

**}**

i**--;**

**}**

**if(** tecla **==** 13 **){**

**if(** i **==** 21 **){**

system**(**"cls"**);**

ObjJuego**.**juegoTetris**();** // metodo que comienza el juego

**}**

**if(** i **==** 22 **){**

system**(**"cls"**);**

system**(**"help.chm"**);**

system**(**"pause"**);**

system**(**"cls"**);**

**}**

**if(** i **==** 23 **){**

system**(**"cls"**);**

ObjJuego**.**impresion**();**

system**(**"pause"**);**

system**(**"cls"**);**

**}**

**if(** i **==** 24 **){**

system**(**"cls"**);**

cout**<<**"\n\n\n||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||"**;**

cout**<<**"\n|||| ||||\n"**;**

cout**<<** "|||| ||||"**;**

cout**<<**"\n|||| GRACIAS POR USAR ESTE PROGRAMA ||||"**;**

cout**<<**"\n|||| ESPERAMOS QUE HAYA SIDO DE TU AGRADO ||||"**;**

cout**<<**"\n|||| ||||"**;**

cout**<<**"\n|||| ||||"**;**

cout**<<**"\n||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||\n\n\n"**;**

system**(**"pause"**);**

exit**(**0**);**

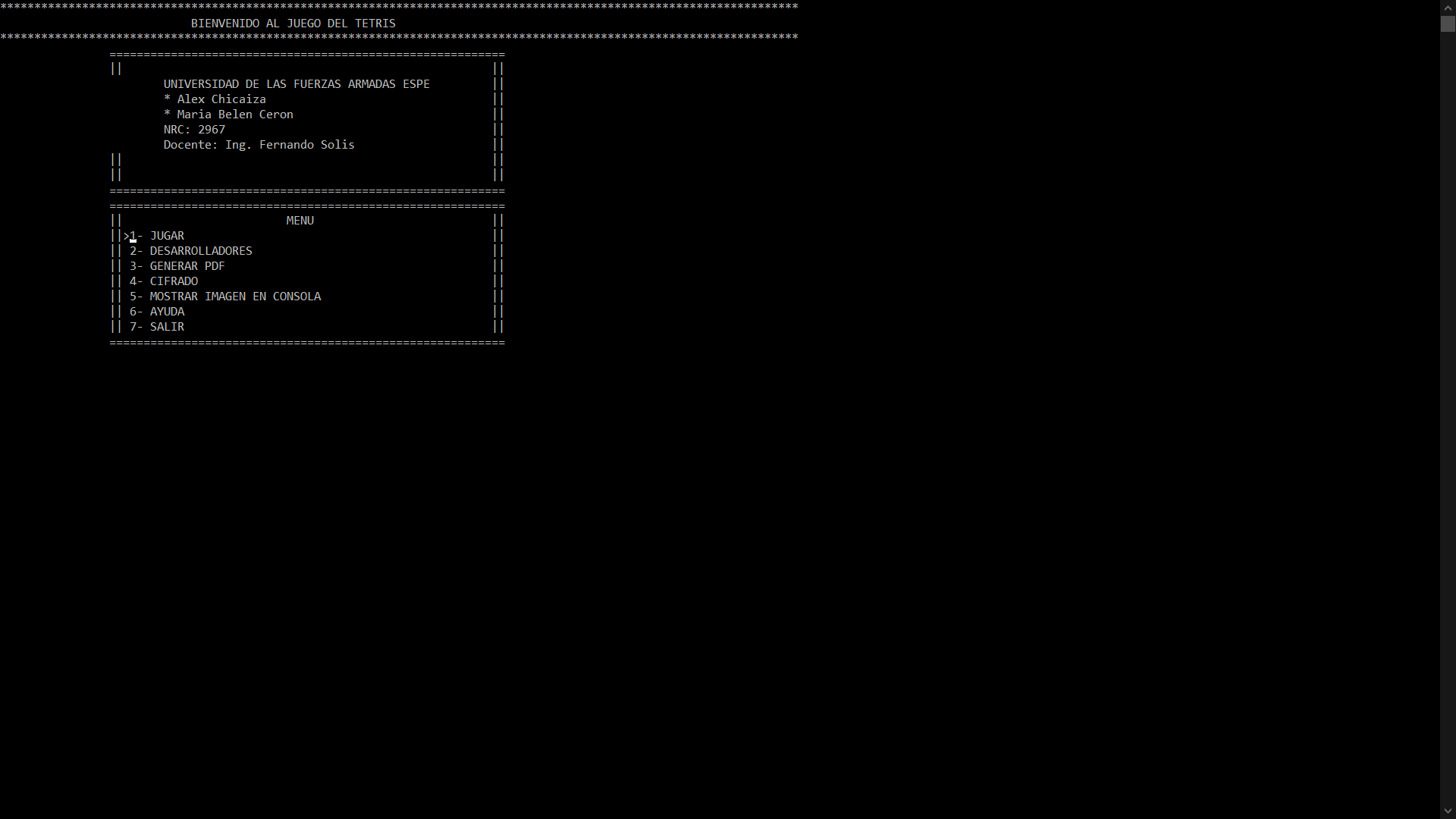
**}**

**}**

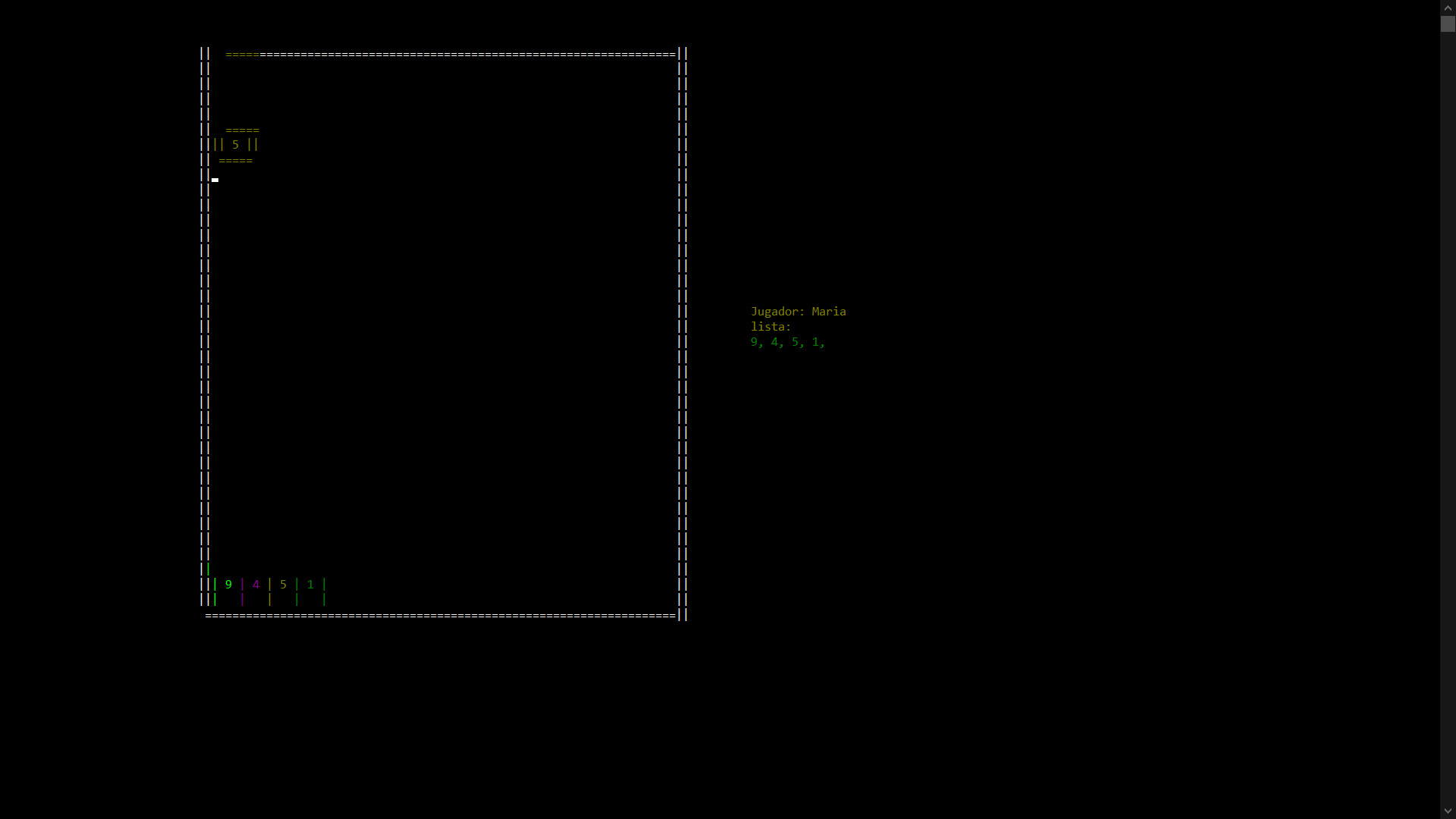
gotoxy**(**30**,** i**);** cout**<<**">"**;**

**}**

1. **Ejecución**







1. **Conclusiones**

Después de las investigaciones realizadas, se puede concluir que el uso de listas doblemente circulares, resultan ser muy útiles al momento de resolver algún problema, además que en la vida real, estas son muy utilizadas por la comunidad en general, nos ayudan a la búsqueda, inserción y eliminación de datos de manera más fácil y eficiente.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DOCENTE RESPONSABLE COORDINADOR DE ÁREA

Ing. Fernando Solis. MsC. PhD. Rodrigo Fonseca.