18-3-2021

Pilas (Torres hanoi)

Practica: 3

Materia: Seminario de estructura de datos 1

Sección: D13.

Código: 216584703

Carrera: Ingeniería en computación.

Nombre alumno: Padilla Pérez Jorge Daray

Nombre profesor: Julio Esteban Valdes Lopez



Índice general

Índice **1**

**Introducción 2**

**Pantallazos 3-6**

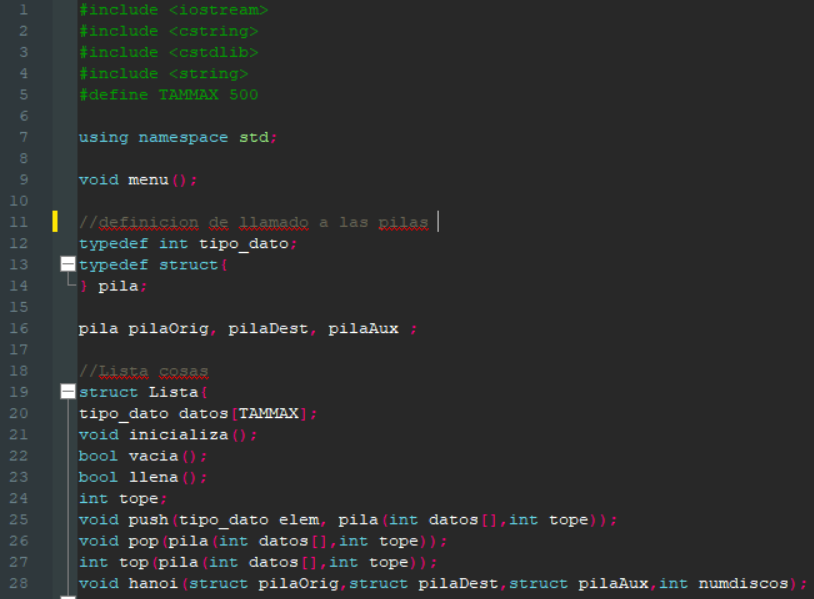
**Conclusión 7**

**Código fuente 8-13**

Introducción

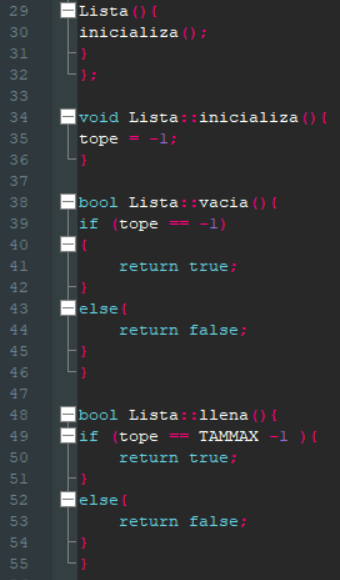
Mi práctica consiste en la implementación De un juego llamado torres de hanoi utilizando pilas en lenguaje c++, en donde se puede apreciar la integración de la frase: divide y vencerás, además de introducir el tema de las pilas y el usar posiciones para agregar y quitar cosas.

Al realizar mi practica obtuve como resultado la incorrecta integración e implementación de las pilas dentro del código, en el cual es necesario asegurar que cada llamada a la pila debe estar definida sobre un problema menos complejo que el que dio lugar a la llamada, como el que yo diseñe, pero a pesar de todo mi esfuerzo no logre completar la practica a tiempo y tampoco funciona correctamente.

Pantallazo

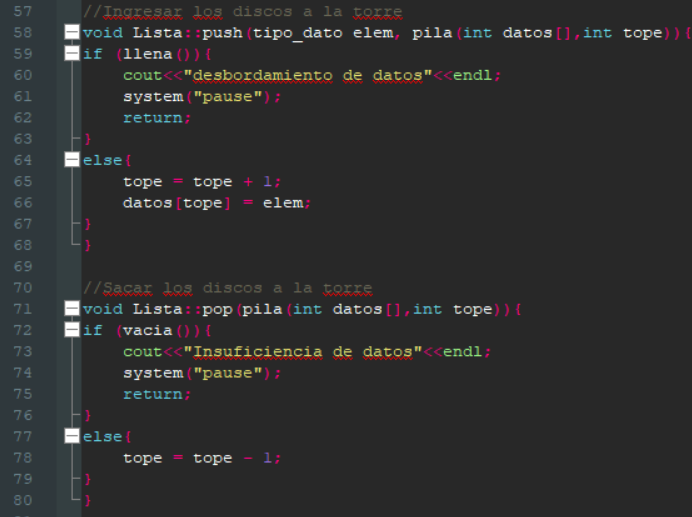
Aquí se intentó la implementación de una función tipo estructura para las pilas a la cual le agregue la pila original, la secundaria y la ultima.

Además de el inicio de las funciones consecutivas para su funcionamiento.

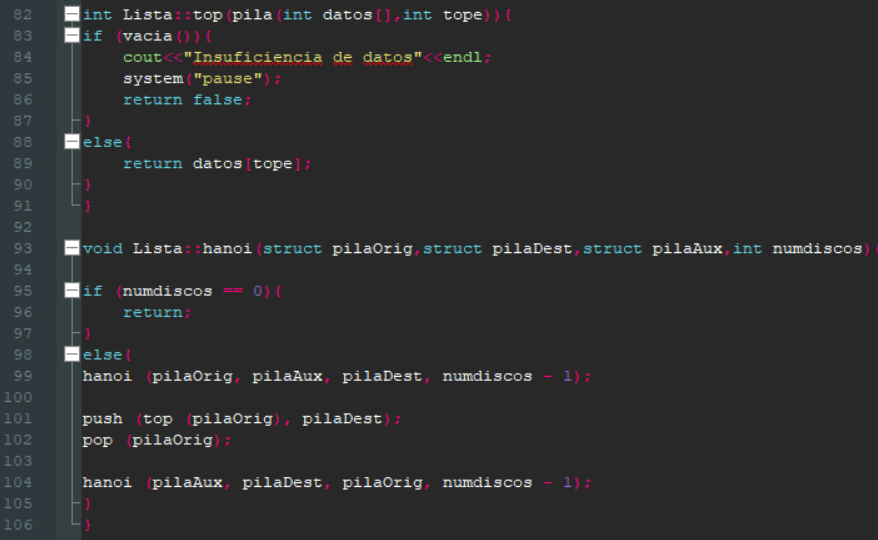


Aquí se inicializa el constructor para la estructura lista y empezamos poniendo las funciones para inicializar, y para medir los errores que pueda generar al jugar el juego.

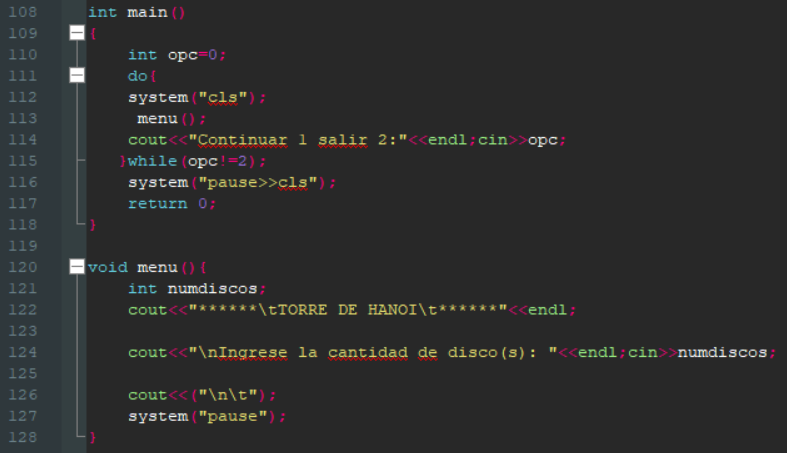
Aquí se mira donde se les dan los parámetros adecuados a las funciones para su correcto funcionamiento.



Aquí se escribe la función push para ingresar los discos, y pop que es para sacar los discos como dice en texto comentado.



Aquí se muestra la función top que sirve para retornar el valor de los datos que estén en el tope de la pila, además de la función hanoi para la correcta implementación de este, mas a parte de aplicar la recursividad mandándose a llamar a si misma la función

aquí se aprecia el menú en la cual se pregunta la cantidad de discos y se almacena en numdiscos para poder contarlos el problema es que me da errores a la hora de implementar estos mismos por lo que no se como solucionarlo pero sigo investigando como hacerle, lastimosamente tengo que entregarlo aunque no funcione para que se pueda evaluar de mejor forma.

Conclusión

Respecto a la realización del código concluyo que está muy difícil todavía para mi y que en este caso que no logro entregarlo completo tengo que seguir estudiando sobre el tema y seguir progresando, teniendo que echarle mas ganas aun de lo que ya hacia para poder concluir bien con el tema de pilas y su correcta implementación.

**Código Fuente**

**#include <iostream>**

**#include <cstring>**

**#include <cstdlib>**

**#include <string>**

**#define TAMMAX 500**

**using namespace std;**

**void menu();**

**//definicion de llamado a las pilas**

**typedef int tipo\_dato;**

**typedef struct{**

**} pila;**

**pila pilaOrig, pilaDest, pilaAux ;**

**//Lista cosas**

**struct Lista{**

**tipo\_dato datos[TAMMAX];**

**void inicializa();**

**bool vacia();**

**bool llena();**

**int tope;**

**void push(tipo\_dato elem, pila(int datos[],int tope));**

**void pop(pila(int datos[],int tope));**

**int top(pila(int datos[],int tope));**

**void hanoi(struct pilaOrig,struct pilaDest,struct pilaAux,int numdiscos);**

**Lista(){**

**inicializa();**

**}**

**};**

**void Lista::inicializa(){**

**tope = -1;**

**}**

**bool Lista::vacia(){**

**if (tope == -1)**

**{**

**return true;**

**}**

**else{**

**return false;**

**}**

**}**

**bool Lista::llena(){**

**if (tope == TAMMAX -1 ){**

**return true;**

**}**

**else{**

**return false;**

**}**

**}**

**//Ingresar los discos a la torre**

**void Lista::push(tipo\_dato elem, pila(int datos[],int tope)){**

**if (llena()){**

**cout<<"desbordamiento de datos"<<endl;**

**system("pause");**

**return;**

**}**

**else{**

**tope = tope + 1;**

**datos[tope] = elem;**

**}**

**}**

**//Sacar los discos a la torre**

**void Lista::pop(pila(int datos[],int tope)){**

**if (vacia()){**

**cout<<"Insuficiencia de datos"<<endl;**

**system("pause");**

**return;**

**}**

**else{**

**tope = tope - 1;**

**}**

**}**

**int Lista::top(pila(int datos[],int tope)){**

**if (vacia()){**

**cout<<"Insuficiencia de datos"<<endl;**

**system("pause");**

**return false;**

**}**

**else{**

**return datos[tope];**

**}**

**}**

**void Lista::hanoi(struct pilaOrig,struct pilaDest,struct pilaAux,int numdiscos){**

**if (numdiscos == 0){**

**return;**

**}**

**else{**

**hanoi (pilaOrig, pilaAux, pilaDest, numdiscos - 1);**

**push (top (pilaOrig), pilaDest);**

**pop (pilaOrig);**

**hanoi (pilaAux, pilaDest, pilaOrig, numdiscos - 1);**

**}**

**}**

**int main()**

**{**

**int opc=0;**

**do{**

**system("cls");**

**menu();**

**cout<<"Continuar 1 salir 2:"<<endl;cin>>opc;**

**}while(opc!=2);**

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

**return 0;**

**}**

**void menu(){**

**int numdiscos;**

**cout<<"\*\*\*\*\*\*\tTORRE DE HANOI\t\*\*\*\*\*\*"<<endl;**

**cout<<"\nIngrese la cantidad de disco(s): "<<endl;cin>>numdiscos;**

**cout<<("\n\t");**

**system("pause");**

**}**