6-9-2021

Estructuras anidadas

Practica: 1

Materia: Estructura de datos

Sección: D01.

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**

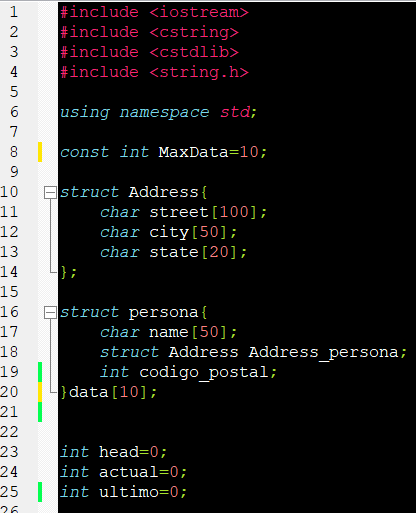
**Código fuente 8…**

Introducción

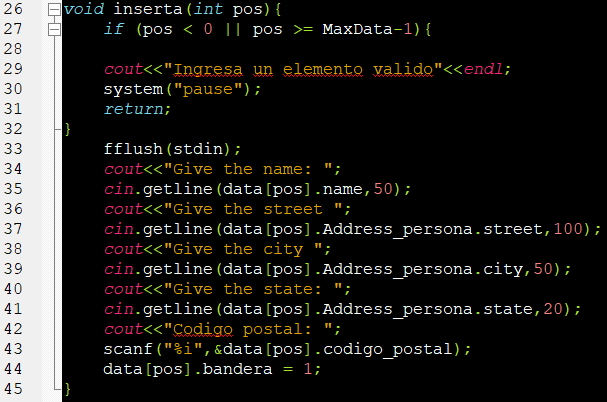
Mi práctica consiste en la implementación de una anidación de estructuras anidadas, yo lo realice en lenguaje c++, en el cual aparte de implementar la anidación de las estructuras, se implementa con registros para tener un control de los datos ingresados por el usuario, en este caso se hizo una estructura de dirección, y otra de persona en la cual esta anidada la dirección de esta persona con un limite de registro de 10 personas.

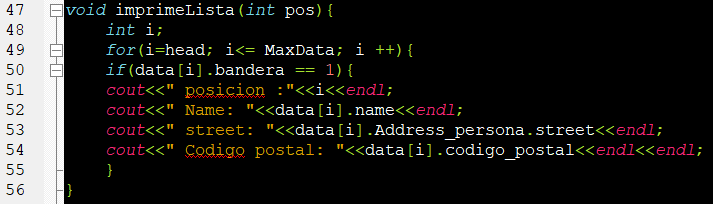
En el mismo programa se incluyen funciones especificas que serian para insertar los elementos, para imprimir estos mismos, para buscar los datos por su posición, y una función para eliminar el nodo seleccionado.

Pantallazos

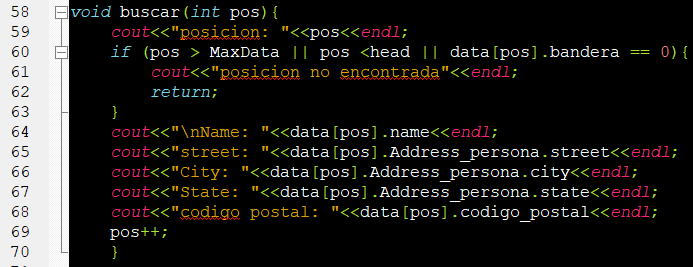
En este pedazo de código se aprecia la estructura address con sus atributos, y debajo de esta la estructura persona la cual tiene sus atributos y la anidacion de la estructura address, con un registro de 10 personas y se inicializa la cabeza en 0.

Aquí se aprecia la función insertar la cual sirve para insertar los elementos que se le piden al usuario, y se terminan guardando en la posición que ingresa el usuario, además de poner la bandera en 1 para saber que está ocupado la posición.

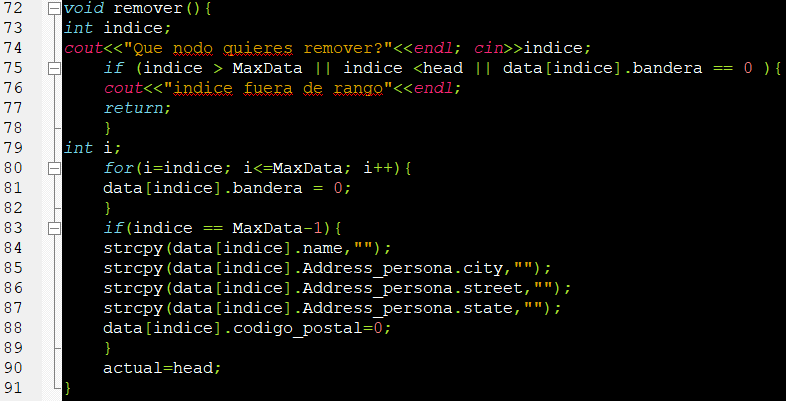


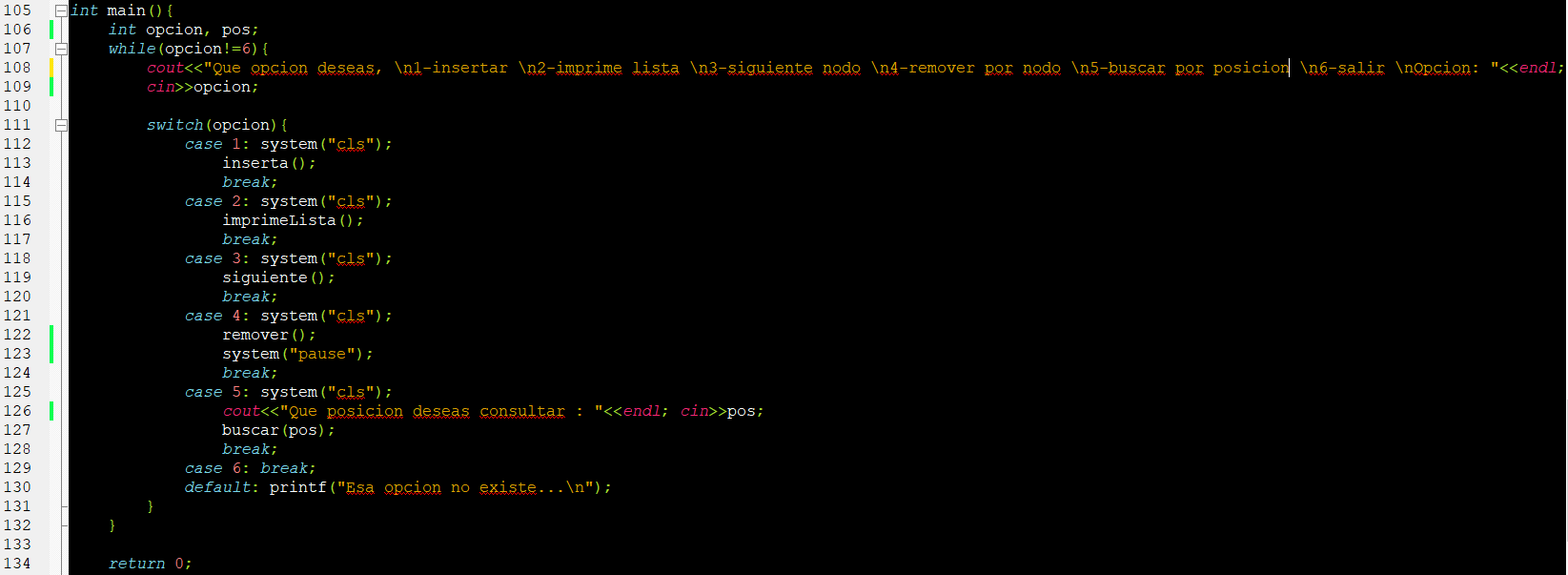
aquí se aprecia la función imprimir, la cual se encargan de que cada vez que la bandera este en 1 imprima los datos más importantes del usuario en cuestión, así imprimiendo solo los nodos que estén ocupados.

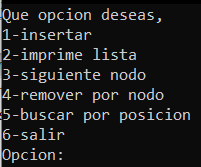
aquí la función buscar tiene de parámetro la posición de este, se evalúa primero si esa posición existe y si esta existe imprime los contenidos de dicha posición, si la posición no existe manda mensaje de no encontrada y retorna al menú principal.

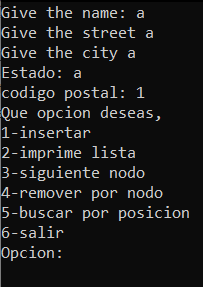


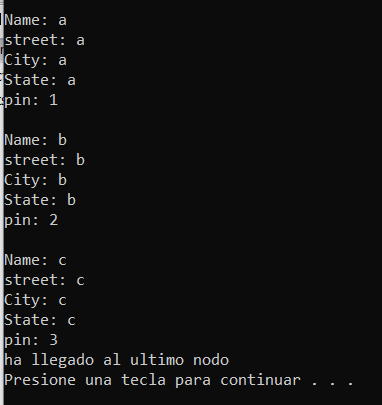
Aquí se implementa la función remover la cual primero pregunta que índice remover y se compruba que este en el rango, después se hace un for el cual nos permite copiar los valores sumándoles uno al índice, y decrementando el valor de ultimo al ser removido, también actual se iguala a la cabeza.



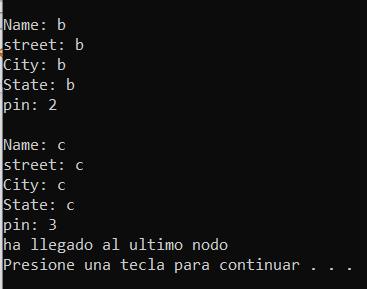
aquí se aprecia el menú.

Este es el menú del usuario

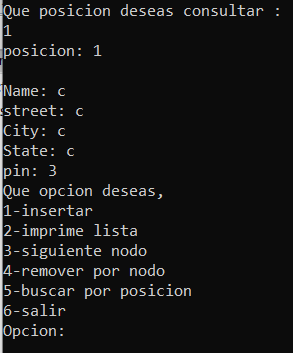
Se insertan los elementos una vez hechos te dirige al menú.



En esta otra se imprime todo.



En este se elimina el primer nodo



Aquí por ultimo se aprecia que al buscar por posición se busca la posición 1 lo cual muestra el segundo nodo porque las posiciones en un arreglo son de 0 en adelante.

Conclusión

Respecto a la realización del código concluyo que tarde mas de lo esperado poder realizarlo ya que no contaba con la información correcta del mismo por haber faltado a la clase en la que se dejo la actividad, sin embargo, gracias a mis compañeros pude llegar a determinar como era el programa solicitado, y su realización.

Un aspecto negativo es que no pude hacer que al momento de llenarse el registro este se vaciara y comenzara de nuevo, intente muchas veces, pero no se logro el cometido, espero que en la clase se puedo solucionar este tema.

**#include <iostream>**

**#include <cstring>**

**#include <cstdlib>**

**#include <string.h>**

**using namespace std;**

**const int MaxData=10;**

**struct Address{**

**char street[100];**

**char city[50];**

**char state[20];**

**};**

**struct persona{**

**char name[50];**

**struct Address Address\_persona;**

**int codigo\_postal;**

**int bandera = 0;**

**}data[10];**

**int head=0;**

**int actual=0;**

**void inserta(int pos){**

**if (pos < 0 || pos >= MaxData-1){**

**cout<<"Ingresa un elemento valido"<<endl;**

**system("pause");**

**return;**

**}**

**fflush(stdin);**

**cout<<"Give the name: ";**

**cin.getline(data[pos].name,50);**

**cout<<"Give the street ";**

**cin.getline(data[pos].Address\_persona.street,100);**

**cout<<"Give the city ";**

**cin.getline(data[pos].Address\_persona.city,50);**

**cout<<"Give the state: ";**

**cin.getline(data[pos].Address\_persona.state,20);**

**cout<<"Codigo postal: ";**

**scanf("%i",&data[pos].codigo\_postal);**

**data[pos].bandera = 1;**

**}**

**void imprimeLista(int pos){**

**int i;**

**for(i=head; i<= MaxData; i ++){**

**if(data[i].bandera == 1){**

**cout<<" posicion :"<<i<<endl;**

**cout<<" Name: "<<data[i].name<<endl;**

**cout<<" street: "<<data[i].Address\_persona.street<<endl;**

**cout<<" Codigo postal: "<<data[i].codigo\_postal<<endl<<endl;**

**}**

**}**

**}**

**void buscar(int pos){**

**cout<<"posicion: "<<pos<<endl;**

**if (pos > MaxData || pos <head || data[pos].bandera == 0){**

**cout<<"posicion no encontrada"<<endl;**

**return;**

**}**

**cout<<"\nName: "<<data[pos].name<<endl;**

**cout<<"street: "<<data[pos].Address\_persona.street<<endl;**

**cout<<"City: "<<data[pos].Address\_persona.city<<endl;**

**cout<<"State: "<<data[pos].Address\_persona.state<<endl;**

**cout<<"codigo postal: "<<data[pos].codigo\_postal<<endl;**

**pos++;**

**}**

**void remover(){**

**int indice;**

**cout<<"Que nodo quieres remover?"<<endl; cin>>indice;**

**if (indice > MaxData || indice <head || data[indice].bandera == 0 ){**

**cout<<"indice fuera de rango"<<endl;**

**return;**

**}**

**int i;**

**for(i=indice; i<=MaxData; i++){**

**data[indice].bandera = 0;**

**}**

**if(indice == MaxData-1){**

**strcpy(data[indice].name,"");**

**strcpy(data[indice].Address\_persona.city,"");**

**strcpy(data[indice].Address\_persona.street,"");**

**strcpy(data[indice].Address\_persona.state,"");**

**data[indice].codigo\_postal=0;**

**}**

**actual=head;**

**}**

**int main(){**

**int opcion, pos;**

**while(opcion!=5){**

**cout<<"Que opcion deseas, \n1-insertar por posicion \n2-imprime lista \n3-remover por posicion \n4-buscar por posicion \n5-salir \nOpcion: "<<endl;**

**cin>>opcion;**

**switch(opcion){**

**case 1: system("cls");**

**cout<<"En que posicion desea insertar el elemento :"<<endl;cin>>pos;**

**inserta(pos);**

**break;**

**case 2: system("cls");**

**imprimeLista(pos);**

**break;**

**case 3: system("cls");**

**remover();**

**system("pause");**

**break;**

**case 4: system("cls");**

**cout<<"Que posicion deseas consultar : "<<endl; cin>>pos;**

**buscar(pos);**

**break;**

**case 5: break;**

**default: printf("Esa opcion no existe...\n");**

**}**

**}**

**return 0;**

**}**