#include <stdio.h>

#include <windows.h>

#include <stdlib.h>

#include <iostream>

using namespace std;//открывает пространство имен библиотеки iostream

typedef struct patients {

char Name[20];

char LastName[20];

char Patronymic[20];

char Address[256];

char HealthCardNumber[20];

char InsuranceNumber[20];

struct patients\* next;

//объявляем шаблон структуры

} Patients;

Patients\* list = NULL;

/\*ОБЪЯВЛЕНИЕ ПРОТОТИПОВ ФУНКЦИЙ\*/

int Menu();

void Input(void);

void Output(void);

void Del();

void CountList();

//void FindMax();//////////////////////////

//void FindMin();///////////////////

void DeleteDouble();/////////////////////

void DeleteList();

void clear\_stream();

void clear\_stream()

{

while (getchar() != '\n');

}

int main()

{

while (1)

{

switch (Menu())//вызов функции menu

{

case 1: Input(); break;//вызов функции ввода

case 2: Output(); break;//вызов функции сортировки

case 3: Del(); break;//вызов функции поиска

case 4: DeleteList(); break;

case 5: CountList(); break;

case 6: DeleteDouble(); break;

case 7: return 0;

}

puts("\n----------------------");

}

}

int Menu() //определение функции menu

{

int choice;

do {

printf\_s("\n Menu \n\n");

printf\_s("1. Input an element of a list; \n"); //Ввод массива структур

printf\_s("2. Output the list; \n"); //Сортировка массива структур

printf\_s("3. Delete an element of the list;\n"); //Поиск

printf\_s("4. Delete the list;\n");

printf\_s("5. Count all the elements of the list;\n");

printf\_s("6. Delete double elements;\n");

printf\_s("7. Exit;\n");

//Изменение заданной структуры

printf\_s("\n Your choise: ");

scanf\_s("%d", &choice);

puts("\n----------------------\n");

} while (choice > 7);

return choice;

}

void Input()//определение ф-ции ввода новой структуры все ок

{

Patients\* p, \* previous, \* current;

p = (Patients\*)malloc(sizeof(Patients));

printf\_s("--- New Patient-- \n"); //выводим номер записи

printf\_s("Input Name:\n");

clear\_stream();

gets\_s(p->Name);

printf\_s("Input Last name:\n");

//clear\_stream();

gets\_s(p->LastName);

printf\_s("Input Patronymic:\n");

//clear\_stream();

gets\_s(p->Patronymic);

printf\_s("Input Address:\n");

//clear\_stream();

gets\_s(p->Address);

printf\_s("Input Health card number:\n");

//clear\_stream();

gets\_s(p->HealthCardNumber);

printf\_s("Insurance number\n");

//clear\_stream();

gets\_s(p->InsuranceNumber);

printf\_s("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

if (p != NULL)

{

p->next = NULL;

Patients\* previous = NULL;

Patients\* current = list;

//while (current != NULL && (p->LastName > current->LastName))

while (current != NULL && (\_stricmp(p->LastName, current->LastName) > 0))

{

previous = current;

current = current->next;

}

if (previous == NULL)

{

p->next = list;

list = p;

}

else

{

previous->next = p;

p->next = current;

}

}

else

printf("No free space!!\n");

}

void Output()//функция вывода массива структур

{

Patients\* p = list;

if (p == NULL)

printf("List is empty\n");

else

{

printf("Your list:\n");

printf\_s("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

int num = 1;

while (p != NULL)

{

printf("Patient Number ");

cout << num;

printf("\n");

printf("Name: ");

printf(p->Name);

printf("\n");

printf("LastName: ");

printf(p->LastName);

printf("\n");

printf("Patronymic: ");

printf(p->Patronymic);

printf("\n");

printf("Address: ");

printf(p->Address);

printf("\n");

printf("HealthCardNumber: ");

printf(p->HealthCardNumber);

printf("\n");

printf("InsuranceNumber: ");

printf(p->InsuranceNumber);

printf("\n");

p = p->next;

num++;

printf\_s("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

}

}

}

void Del()

{

Patients\* prev = list, \* cur = list->next, \* p;

char value[20];

printf\_s("Input Last name of Patient you want to delete:\n");

clear\_stream();

gets\_s(value);

if (strcmp(value, list->LastName) == 0) { /\* если нужно удалить \*/

/\* начальный элемент списка \*/

p = list; /\* p указывает на удаляемый элемент\*/

list = list->next; /\* list указывает на след. элемент \*/

free(p); /\* удалить элемент \*/

printf\_s("Element is deleted\n");

}

else { /\* иначе... \*/

/\* С помощью цикла while найти элемент \*/

while (cur != NULL && strcmp(value, cur->LastName) != 0) {

prev = cur;

cur = cur->next;

}

if (cur != NULL) { /\* если элемент найден... \*/

p = cur; /\* p указывает на удаляемый элемент\*/

prev->next = cur->next; /\* перенастроить связи \*/

free(p); /\* удалить элемент \*/

printf\_s("Element is deleted\n");

}

else {

printf\_s("There is no such an element\n");

}

}

}

void DeleteList()

{

Patients\* prev = list, \* p;

if (list != NULL)

{

while (list != NULL)

{

p = list; /\* p указывает на удаляемый элемент\*/

list = list->next; /\* list указывает на след. элемент \*/

free(p); /\* удалить элемент \*/

printf\_s("Element has been deleted\n");

}

printf\_s("The list has been deleted");// почему не пишет что список удален?????

}

else if (list = NULL)

{

printf\_s("The list is empty");// почему не пишет что список удален?????

}

}

void CountList()//все ок

{

int count = 0;

Patients\* prev = list, \* p;

p = list;

if (p != NULL)

{

while (p != NULL)

{

p = p->next;

count++;

}

printf\_s("The list has %d elements", count);

}

else {

printf\_s("The list is empty");

}

}

void DeleteDouble()

{

}