

Міністерство освіти і науки України
Національний університет “Львівська політехніка”
Кафедра систем штучного інтелекту



ЗВІТ
про виконання лабораторної роботи №9
з дисципліни “Алгоритмізація та програмування, частина 1”

Виконав:

студент групи КН-110

Дойков В.С.

Викладач:

Гасько Р.Т

2018

Варіант №9

Тема: "Блоковий ввід-вивід"

Мета роботи: Робота із двійковими файлами, організація вводу-виводу структурованої інформації і її зберігання на зовнішніх носіях.

Постановка завдання:

Структура "Пацієнт":

- прізвище, ім'я, по батькові;
- домашня адреса;
- номер медичної карти;
- номер страхового поліса.

Знищити елемент із заданим номером медичної карти, додати 2 елементи в початок файлу.

Результат обчислення:

```
jharvard@appliance (~): ./lab8
Amount of patients is 2
Name #1 is Vadim
Adress #1 is 12 street
Medical ID #1 is 1212
Polis number #1 is 2323
Name #2 is Igor
Adress #2 is 13street
Medical ID #2 is 4545
Polis number #2 is 5667
The Medical ID to find is 4545
DELETED
New name is Igor
New adress is 2323
New Medical ID is 2345
```

Код програми:

```
#include <cs50.h>
#include <stdio.h>
#include <stdlib.h>

int amnt;

typedef struct{
    char* name;
    char* adress;
    char* med;
    char* polis;
} PATIENTS;
```

```
//PROTOTYPES OF FUNCTIONS////////////////////////////////////
```

```
void WRITE(PATIENTS c, int amnt);  
void READ(PATIENTS re[amnt]);  
void DELETE(PATIENTS re[amnt],int key);  
void FIND(PATIENTS re[amnt], char com[25]);  
void ADD(PATIENTS re[amnt]);  
void SAVE(PATIENTS re[amnt]);
```

```
//MAIN////////////////////////////////////
```

```
int main(){  
PATIENTS c;  
printf("Amount of patients is ");  
int amnt = GetInt();  
PATIENTS re[amnt+1];  
WRITE(c, amnt);  
READ(re);  
printf("The Medical ID to find is ");  
char com[25];  
scanf("%s", com);  
FIND(re, com);  
ADD(re);  
SAVE(re);  
}
```

```
void WRITE(PATIENTS c, int amnt){  
FILE *f;  
if ((f = fopen("f.bin", "wb")) == NULL)exit(1);  
for (int i = 1; i <= amnt; i++){  
printf("Name #%i is ", i); c.name=GetString();  
printf("Adress #%i is ", i); c.adress=GetString();  
printf("Medical ID #%i is ", i); c.med=GetString();  
printf("Polis number #%i is ", i); c.polis=GetString();  
fwrite(&c, sizeof(PATIENTS), 1, f);  
}  
fclose(f);  
}
```

```
void READ(PATIENTS re[amnt]){  
FILE *f;  
if ((f = fopen("f.bin", "rb")) == NULL) exit(1);  
int i = 0;
```

```

while (!feof(f) && (i < amnt)){
fread(&re[i], sizeof(PATIENTS), 1, f);
i++;
}
fclose(f);
}

```

```

void DELETE(PATIENTS re[amnt], int key){
    for(int i=key; i<amnt; i++){
        re[i]=re[i+1];
    }
    amnt--;
}

```

```

void FIND(PATIENTS re[amnt], char com[25]){
    int cur;
    bool equal;
    for (cur = 0; cur < amnt; cur++){
        equal = 1;
        for (int j = 0; com[j] != '\0'; j++) if(com[j] != re[cur].med[j]) equal = 0;
        if (equal) break;
    }
    if (equal) DELETE (re, cur);
    printf("DELETED\n");
}

```

```

void ADD(PATIENTS re[amnt]){
    for (int i = 3; i < amnt+1; i++) re[i+2] = re[i];
    for (int j = 1; j <=2; j++){
        printf("New name is "); scanf("%s", re[j].name);
        printf("New adress is "); scanf("%s", re[j].adress);
        printf("New Medical ID is "); scanf("%s", re[j].med);
        printf("New polis is "); scanf("%s", re[j].polis);
        amnt++;
    }
}

```

```

void SAVE(PATIENTS re[amnt]){

```

```
FILE *f;

if ((f = fopen("f.bin", "w")) == NULL) exit(1);

for (int i = 0; i < amnt; i++) fwrite(&re[i], sizeof(PATIENTS), 1, f);

fclose(f);

printf("\n\x1b[32mChanges have been saved\x1b[0m\n\n");

}
```