Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут»

Кафедра прикладної математики
(назва кафедри)

Лабораторна робота № 8б

з дисципліни «Програмування»

Виконав:

Студент групи КМ-53 Галета Максим Сергійович _____
(шифр групи) (прізвище, ім'я, по батькові) (підпис)

Перевірив:

Викладач Громова В. В. _____ **Оцінка** _____
(прізвище, ініціали) (підпис)

Київ – 2016

ЗМІСТ

1 МЕТА РОБОТИ

Вивчення файлових типів даних, придбання практичних навичок створення й редагування текстових і бінарних файлів у мові С

2 ЗАВДАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Скласти програму на мові С, яка робить обчислення відповідно до варіанта.

Варіант №4

1. Структура "Пацієнт":
   - прізвище ім'я по батькові;
   - Домашня адреса;
   - Номер медичної карти;
   - Номер страхового поліса;
   - Статус запису.
2. Видалити елемент із заданим номером медичної карти,
3. Додати 2 елементи в початок файлу.

3 ВИМОГИ ДО ПРОГРАМИ

1. Для запису структури в файл і зчитування структури з файлу використовувати функції блочного введення/виведення fread і fwrite.

2. Для видалення/додавання елементів в файл використовувати допоміжний файл.

3.Запис не повинен видалятися негайно з файлу, а тільки позначатися як видалений (у вікні для перегляду повинні відображатися або активні, не видалені записи, або помічені як видалені, але не обидва типи записів одночасно). Поле структури статус запису може містити 1 або 0. При створенні файлу структур в це поле треба записати - 0 (ознака активного запису), а при видаленні запису з файлу - записати 1.

4. Перегляд записів - перегортання вперед, назад, в кінець файлу або в його початок.

5. Повинна бути передбачена можливість відновлення видаленого запису зі списку видалених (але ця можливість забезпечується до моменту отримання ущільненого файлу - див. Далі).

6. Збереження файлу з даними повинно виконуватися в 2-х режимах - з ущільненням і без ущільнення (в першому випадку записи, помічені як видалені, викидаються, а в другому випадку - в файл записуються всі записи зі збереженням їх статусу).

7. Взаємодія з файлом даних повинне здійснюватися в бінарному режимі, тобто записи зберігаються як їх образ в основній пам'яті комп'ютера.

4 СКЛАД І СТРУКТУРА ДАНИХ

В області оголошення змінних задані такі змінні:

Глобальні змінні: patient *head_main, *first_main, *head_accesory; int count1, count2.

1. Функція **void menu()**:
   int punkt, ch;
2. Функція **patient *new_patient()**
   int y;
   patient *element;
3. Функція **void add_new_patient(patient *element)**
   patient *current;
4. Функція **void save()**
   patient *current;
5. Функція **void load()**
   patient *element, *current, *prev;
6. Функція **void list(patient *pointer)**
   int counter;
   patient *temp;
7. Функція **void delete_patient()**
   char s[40];
   int y, number;
8. Функція **void recovery_patient()**
   char s[40];
   int y, number;

5 ОПИС ПРОГРАМИ

Функція **void frame()** являє собою обкладинку програми. На екран виводяться: номер лабораторної роботи, тема, варіант, ініціали студента.

Функція **void menu()** представляє собою меню програми.

Функція **patient *new_patient()** являє собою введення користувачем даних пацієнта.

Функція **void add_new_patient(patient *element)** додає нового пацієнта у список.

Функція **void save()** зберігає інформацію в файл.

Функція **void load()** завантажує інформацію з файлу.

Функція **void delete_patient()** видаляє пацієнта зі списку.

Функція **void list()** виводить весь список на екран.

Функція **void recovery_patient()** відновлює пацієнта назад у список.

ВИСНОВКИ

Під час виконання цієї лабораторної роботи вивчено файлові типи даних, придбано практичні навички створення й редагування текстових і бінарних файлів умові С.

Додаток А. Текст програми

```
void menu()
{
	int punkt=0, ch;
	const char *PunktMenu[9]=
	{
		" Завантажити		",
		" Додати нового пацієнта	",
		" Зберегти		",
		" Список пацієнтів		",
		" Пошук пацієнтів		",
		" Видалити пацієнта		",
		" Відновити пацієнта		",
		" Список видалених пацієнтів",
		" Вийти з програми		",
	};
	while(true)
	{
		ukr(1251);
		textcolor(LightCyan);
		gotoxy(37,1);
		printf("МЕНЮ");
		gotoxy(37,2);
		ukr(866);
		for(short i=0; i<4; i++)
			putchar(196);
		ukr(1251);
		for(int i=0; i<9; i++)
		{
			if (i==punkt)
			{
				textcolor(Black);
				textbackground(LightCyan);
			gotoxy(2,4+i);
				printf(“%s”,PunktMenu[i]);
				textbackground(Black);
		}
		else
		{
			textbackground(Black);
			textcolor(White);
```

```
                gotoxy(2,4+i);
                printf("%s",PunktMenu[i]);
        }
}
gotoxy(79,13);
ch=getch();
switch(ch)
{
        case 80:
                punkt+=1;
                if(punkt>8)
                        punkt=0;
                break;
        case 72:
                punkt-=1;
                if(punkt<0)
                        punkt=8;
                break;
        case 13:
                if(punkt==0)
                        load();
                if(punkt==1)
                        add_new_patient(new_patient());
                if(punkt==2)
                        save();
                if(punkt==3)
                {
                        system("cls");
                        choose_sort();
                }
                if(punkt==4)
                {
                        system("cls");
                        choose_search();
                }
                if(punkt==5)
                        delete_patient();
                if(punkt==6)
                        recovery_patient();
                if(punkt==7)
                        list(head_accesory);
                if(punkt==8)
                {
```

```
                                        free_memory_head_main();
                                        free_memory_head_accesory();
                                        return;
                                }
                                break;
                        }
                }
}


void choose_sort()
{
        int punkt=0, ch;
        string PunktMenu[3]=
        {
                " Впорядкувати пацієнтів за ПІБ                ",
                " Впорядкувати пацієнтів за номером медичної картки",
                " Їповернутись до головного меню                "
        };
        while(true)
        {
                textcolor(LightCyan);
                gotoxy(31,1);
                printf("Список пацієнтів");
                gotoxy(31,2);
                ukr(866);
                for(short i=0; i<16; i++)
                        putchar(196);
                ukr(1251);
                for(int i=0; i<3; i++)
                {
                        if (i==punkt)
                        {
                                textcolor(Black);
                                textbackground(LightCyan);
                        gotoxy(2,4+i);
                                printf("%s",PunktMenu[i]);
                                textbackground(Black);
                }
                else
                {
                        textbackground(Black);
                        textcolor(White);
```

```
                        gotoxy(2,4+i);
                        printf("%s",PunktMenu[i]);
                }
        }
        gotoxy(79,13);
        ch=getch();
        switch(ch)
        {
                case 80:
                        punkt+=1;
                        if(punkt>2)
                                punkt=0;
                        break;
                case 72:
                        punkt-=1;
                        if(punkt<0)
                                punkt=2;
                        break;
                case 13:
                        if(punkt==0)
                                sort_pib();
                        if(punkt==1)
                                sort_number_card();
                        if(punkt==2)
                        {
                                system("cls");
                                return;
                        }
                        break;
        }
    }
}


void choose_search()
{
    int punkt=0, ch;
    string PunktMenu[3]=
    {
        " Пошук пацієнтів за ПІБ                ",
        " Пошук пацієнтів за номером медичної картки ",
        " Повернутись до головного меню          "
    };
```

```
while(true)
{
	textcolor(LightCyan);
	gotoxy(31,1);
	printf("Пошук пацієнтів);
	gotoxy(31,2);
	ukr(866);
	for(short i=0; i<15; i++)
		putchar(196);
	ukr(1251);
	for(int i=0; i<3; i++)
	{
		if (i==punkt)
		{
			textcolor(Black);
			textbackground(LightCyan);
		gotoxy(2,4+i);
			printf(“%s”,PunktMenu[i]);
			textbackground(Black);
	}
	else
	{
		textbackground(Black);
		textcolor(White);
			gotoxy(2,4+i);
			printf(“%s”,PunktMenu[i]);
		}
	}
	textcolor(White);
	gotoxy(79,13);
	ch=getch();
	switch(ch)
	{
		case 80:
			punkt+=1;
			if(punkt>2)
				punkt=0;
			break;
		case 72:
			punkt-=1;
			if(punkt<0)
				punkt=2;
			break;
```

```
                    case 13:
                        if(punkt==0)
                            search_pib();
                        if(punkt==1)
                            search_number_card();
                        if(punkt==2)
                        {
                            system("cls");
                            return;
                        }
                        break;
                }
            }
        }
```