# НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи №*4*

із дисципліни «Програмування»

на тему

*«Вказівники, функція, рекурсія»*

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-63 | *асистент Громова В. В.* |
| *Артеменко Я.К.* |  |

Київ — 2017

# **ЗМІСТ**

[1 ПОСТАНОВКА ЗАДАЧІ 2](#_Toc478382182)

[1.1 Мета роботи 2](#_Toc478382183)

[1.2 Завдання на лабораторну роботу 2](#_Toc478382184)

[2 ОПИС ПРОГРАМИ 3](#_Toc478382185)

[ВИСНОВКИ 4](#_Toc478382186)

[Додаток А (блок-схема) 4](#_Toc478382187)

[Додаток Б (відповіді на контрольні запитання) 6](#_Toc478382188)

[Додаток В (текст програми) 9](#_Toc478382189)

# **1 ПОСТАНОВКА ЗАДАЧІ**

* 1. Мета роботи

Вивчити синтаксис і правила роботи з типами даних, визначеними користувачем.

* 1. Завдання на лабораторну роботу

Варіант №13

1. Описати структуру з іменем NOTE, що містить такі поля:
2. Прізвище, ім’я;
3. Номер телефону;
4. День народження (масив із трьох чисел);
5. Написати програму, що виконує такі дії:
6. Уведення з клавіатури даних у масив із 5 структур типу NOTE;
7. Упорядкування записів за алфавітом;
8. Виведення на екран інформації про людей, чиї дні народження припадають на місяць, значення якого уведено з клавіатури; якщо таких людей немає, вивести на екран відповідне повідомлення.

**Постановка завдання**

Програма повинна задовольняти такі вимоги:

1. Уведення даних структури з клавіатури.
2. Виконання роботи з масивом структур згідно з вимогами завдання.
3. Виведення результату на екран, якщо потрібно, у вигляді таблиці. Дані всередині структури повинні відповідати темі завдання.

# **ОПИС ПРОГРАМИ**

Дана програма містить 9 функцій: menu\_main, integer, valid\_bit, table, sort, search\_note, input\_note, menu, main. Ще програма містить структуру NOTE із трьома полями(вказано в пункті 1.2) .

Функція **menu\_main** виводить на екран дані про виконавця та надає можливість користувачу обрати завдання для виконання. Обрати завдання можна за допомогою стрілок «вверх» і «вниз» і за допомогою клавіші «Enter» перейти до його виконання. Пункти, які ми можемо обрати підсвічені білим кольором, а умови завдань виділяються сірим кольором. За допомогою оператора розгалуження «if – else» зроблено можливість переміщення «стрілки вибору» між пунктами меню та за допомогою десяткових індексів функціональних клавіш («80» – для стрілки вниз, а «72» – для стрілки вверх) зроблено можливим використання клавіш для вибору. За допомогою циклу «while» очікується натиснення клавіші Enter, після чого ця функція повертає значення вибраного пункту в функцію main.

Функції **integer та valid\_bit** перевіряють правильність введених даних. Якщо буде введена буква, а не цифра, то програма покаже повідомлення про помилку.

Функція **table** задає заголовок таблиці, в яку будуть заноситись введені дані.

Функція **sort** сортує введені дані по алфавіту.

Функція **search\_note** використовується для побудови таблиці з уведених даних та виведення інформації про людей, чиї дні народження припадають на місяць, значення якого уведено з клавіатури. Якщо таких людей немає, то виводиться відповідне повідомлення.

Функція **input\_note** використовується для введення даних в структуру NOTE.

Функція **menu** пропонує користувачу обрати подальші дії після знаходження інформації про людину, такі як: повторення всієї роботи з початку, повторення пошуку інформації по номеру місяця та повернення в початкове меню.

Функція **main** є головною і в ній викликаються всі функції і забезпечується робота меню за допомогу оператора «case».

# **ВИСНОВКИ**

На цій лабораторній роботі здобуто навички по роботі з типами даних, які визначені користувачем – структурами, та набуто практичні навички їх використання для написання програм.

# **Додаток А (блок-схема)**

Функція menu\_main:

Початок

System(“cls”);

Printf(“Інформація про виконавця”)

Printf(“Умова завдання”)

ні

If key = {0}

так

Cout<<”-> Продовжити”

Cout<<” Продовжити”

If key = {1}

ні

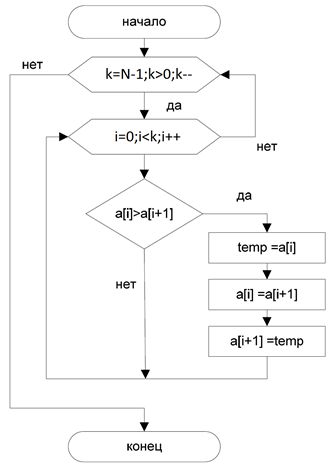
так

Cout<<”-> Вихід”

Cout<<” Вихід”

Кінець

Функція sort:



Функція input\_note:

Початок

Int i=0;i<5;i++

printf("Введите месяц: ");

cin >> number[i].phone\_number;

cin >> number[i].name[1];

printf("Введите фамилию пользователя №%d: ", i + 1);

cin >> number[i].name[0];

printf("Введите имя пользователя №%d: ", i + 1);

printf("Введите номер телефона пользователя №%d: ", i + 1);

printf("Введите день: ");

number[i].data[0] = valid\_bit(31, 1);

number[i].data[1] = valid\_bit(12, 1);

printf("Введите год: ");

number[i].data[2] = valid\_bit(2017, 1940);

# 

Кінець

# **Додаток Б (відповіді на контрольні запитання)**

1. Для ефективного способу подання інформації.
2. Прості типи даних: цілі числа, дійсні числа, логічний, перелічуваний і символьний типи. Складені: структура, об’єднання, перерахування, класи.
3. Структура – елемент, який дає змогу включати в себе дані різних типів, а також і інші структури. Масив – це набір елементів даних одного типу.
4. Способи опису структури:

struct student{

char name[30];

int year;

char group[3];

int scholarship;

};

1. Доступ до конкретного елемента (поля) структури здійснюють за допомогою операції («крапка», dot). Наприклад: strcpy(stud1.name,”Кузина А.А.”).
2. Операції, які допустимі над структурою в цілому: об’єднання структури в масив, друкувати як і окремі елементи структури, так і всю структуру, перейменовувати, та інші.
3. Як формальний параметр можна передати за значенням усю структуру. Також можна створити вказівник на структуру й передавати аргумент типу структури за посиланням.
4. Опис вказівника на структуру:

Struct stru\* adr\_pointer;

де adr\_pointer – змінна-вказівник на структуру struct stru.

1. Як поля структури можна використовувати масиви, структури й масиви структур, наприклад:

Struct addr{char city[34];char street[76];int house;};

struct fulladdr { struct addr address; int room; char name[48]; } a, b;

де addr — шаблон структури, визначений перед оголошенням структури fulladdr та оголошенням змінної а типу структури fulladdr. Для присвоєння значення полю house структури address змінної а потрібно використати a.adress.house = 234.

1. Якщо структуру передають за значенням, то всі поля структури заносять у стек.
2. Можна. Використовуючи вказівник на структуру можна використовувати інший спосіб – використовуючи спеціальну операцію мови С -> («стрілка», arrow).
3. Об’єднання описують так само, як і структури, але замість ключового слова struct використовують ключове слово union. Як і для структур, можна визначити шаблон об’єднання з іменем типу й задати в цьому ж операторі одну чи декілька змінних об’єднання. Або ж опис змінних можна відокремити від визначення шаблону.
4. На відміну від структури, для змінної типу union місця в пам’яті виділяється рівно стільки, скільки потрібно полю об’єднання, що має найбільший розмір у байтах.
5. Синтаксис використання полів об’єднання такий самий, як і для структури.
6. Опис перерахованого типу даних здійснюють оператором виду:

enum *<ім’я\_типу> {<список\_значень>} <список змінних>;*

1. Можна оголошувати й масиви перерахованого типу, наприклад:

enum day week[6];

1. Дає змогу дати нову назву вже існуючому типу даних. Для цього використовують ключове слово typedef. При цьому створюється новий тип даних.

# **Додаток В (текст програми)**

#include <stdio.h>

#include <ctype.h>

#include <math.h>

#include <conio.h>

#include <string.h>

#include <iostream>

#include <locale.h>

#include <windows.h>

using namespace std;

void menu(struct NOTE number[5]);

struct NOTE

{

char name[2][15];

int phone\_number;

int data[3];

};

int menu\_main()

{

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

char key = 0;

int code;

do

{

system("cls");

printf("\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t \* Лабораторная работа №5 \*\n");

printf("\t \* Типы данных, \*\n");

printf("\t \* определенные пользователем \*\n");

printf("\t \* Вариант №13 \*\n");

printf("\t \* Группа KM-63 Артеменко Я.К.\*\n");

printf("\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n\n");

key = (key + 2) % 2;

SetConsoleTextAttribute(hStdOut, (WORD)((0 << 4) | 8));

printf("\tИНСТРУКЦИЯ!!! ДЛЯ НАВИГАЦИИ ПО МЕНЮ ИСПОЛЬЗУЙТЕ СТРЕЛКИ ВВЕРХ, ВНИЗ И КЛАВИШУ ENTER\n");

printf("\t1. Описать структуру с именем NOTE, которая имеет такие поля:\n");

printf("\t - фамилия, имя; \n");

printf("\t - номер телефона;\n");

printf("\t - день рождения (массив из трех чисел);\n\n");

printf("\t2. Написать программу, которая выполняет такие действия:\n");

printf("\t - ввод с клавиатуры данных у массив из 5 структур типа NOTE;\n");

printf("\t - упорядочения записей за алфавитом; \n");

printf("\t - вывод на экран информации о людях, чьи дни рождения \n");

printf("\t приходятся на месяц, значение которого введено с клавиатуры;\n");

printf("\t если таких людей нет, вывести на экран соответствующее сообщение.\n\n\n");

SetConsoleTextAttribute(hStdOut, (WORD)((0 << 4) | 15));

if (key == 0) cout << "\t -> Продолжить. \n" << endl;

else cout << "\t Продолжить.\n" << endl;

if (key == 1) cout << "\t -> Выход.\n" << endl;

else cout << "\t Выход.\n" << endl;

code = \_getch();

if (code == 224)

{

code = \_getch();

if (code == 80) key++;

if (code == 72) key--;

}

} while (code != 13);

system("cls");

return key;

}

int integer(){

char symbol[20];

int z = 0, k, res;

do

{

fflush(stdin); /\*занулення буфера від попередніх записів\*/

scanf("%s", &symbol);

while (symbol[z] >= '0' && symbol[z] <= '9')

z++;

res = atoi(symbol);

if (symbol[z])

printf("Entered symbol is not an integer number! Try again: ");

else

return res;

} while (symbol[z]);

}

int valid\_bit(int a, int b) /\*валідація для бітових операцій на цілі числа (в залежності від потрібних значень)\*/ {

int c;

do

{

c = integer();

if (c < b || c > a)

printf("Too many items! Try again: ");

} while (c < b || c > a);

return c;

}

void table()

{

printf("----------------------------------------------------------------\n");

printf("| #| Name | phone\_number | Data |\n");

printf("----------------------------------------------------------------\n");

}

void sort(struct NOTE number[5])

{

NOTE p;

for (int j = 0; j<5; j++)

{

for (int k = 0; k<5; k++)

{

if (strcmp(number[j].name[0], number[k].name[0])<0)

{

p = number[j];

number[j] = number[k];

number[k] = p;

}

}

}

}

void search\_note(struct NOTE number[5])

{

int month;

system("cls");

cout << "Отсортированный: " << endl;

table();

for (int i = 0; i < 5; i++)

{

printf("|%2.d|%15s %15s|%15d|%2.d/%2.d/%4.d |\n", i + 1, number[i].name[0], number[i].name[1], number[i].phone\_number, number[i].data[0], number[i].data[1], number[i].data[2]);

printf("----------------------------------------------------------------\n");

}

cout << "Введите месяц из даты рождения человека: ";

cin >> month;

bool flag=false;

for (int i = 0; i < 5; i++)

{

if (number[i].data[1] == month)

{

flag = true;

system("cls");

printf("|%2.d|%15s %15s|%15d|%2.d/%2.d/%4.d |\n", i + 1, number[i].name[0], number[i].name[1], number[i].phone\_number, number[i].data[0], number[i].data[1], number[i].data[2]);

printf("----------------------------------------------------------------\n");

}

}

if (!flag)

{

system("cls");

cout << "В базе нету такого человека" << endl;

}

menu(number);

}

void input\_note()

{

struct NOTE number[5];

for (int i = 0; i < 5; i++)

{

printf("Введите фамилию пользователя №%d: ", i + 1);

cin >> number[i].name[0];

printf("Введите имя пользователя №%d: ", i + 1);

cin >> number[i].name[1];

printf("Введите номер телефона пользователя №%d: ", i + 1);

cin >> number[i].phone\_number;

cout << "Введите дату рождения пользователя №" << i + 1 << endl;

printf("Введите день: ");

number[i].data[0] = valid\_bit(31, 1);

printf("Введите месяц: ");

number[i].data[1] = valid\_bit(12, 1);

printf("Введите год: ");

number[i].data[2] = valid\_bit(2017, 1940);

}

cout << "\n\n";

sort(number);

search\_note(number);

system("pause");

}

void menu(struct NOTE number[5])

{

printf("1 - Повторить всю работу;\n");

printf("2 - Повторить поиск;\n");

printf("3 - Вернуться в начальное меню");

char input;

input = \_getch();

switch (input) {

case '1':

system("cls");

input\_note();

break;

case '2':

system("cls");

search\_note(number);

break;

case '3':

system("cls");

menu\_main();

break;

}

}

int main()

{

while (1)

{

setlocale(LC\_ALL, "RUS");

int i1 = menu\_main() + 1;

switch (i1)

{

case 1: {input\_note(); break; }

case 2: {return 0; }

}

}

}