МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЖИТОМИРСЬКИЙ АГРОТЕХНІЧНИЙ КОЛЕДЖ

ВІДДІЛЕННЯ: «Електрифікація та інформаційні системи»

Спеціальність: «Обслуговування програмних систем та комплексів»

Звіт з практики «ТРПЗ»

Виконав: студент групи П-31

Сапожник Д.О.

Перевірив :Устименко Я.І

Житомир 2017

Зміст

[**Лабораторна робота №1 3**](#_Toc476514283)

[**Лабораторна робота №2 7**](#_Toc476514284)

[**Лабораторна робота №3 9**](#_Toc476514285)

[**Лабораторна робота №4 17**](#_Toc476514286)

[**Лабораторна робота №5 24**](#_Toc476514287)

[**Лабораторна робота №6 25**](#_Toc476514288)

[**Лабораторна робота №7 27**](#_Toc476514289)

[**Лабораторна робота №8 31**](#_Toc476514290)

[**Лабораторна робота №9 35**](#_Toc476514291)

[**Лабораторна робота №10 39**](#_Toc476514292)

[**Лабораторна робота №11 42**](#_Toc476514293)

[**Лабораторна робота №12 46**](#_Toc476514294)

[**Лабораторна робота №13 49**](#_Toc476514295)

[**Лабораторна робота №14 52**](#_Toc476514296)

## Лабораторна робота №1

***Тема:***Прості типи даних. Базові конструкції мови С

***Мета:*** навчитися складати алгоритми розв’язку задач у вигляді блок-схем, ознайомитися з простими типами даних та базовими конструкціями мови Сі, оволодіти практичними навичками складання, введення, редагування і виконання найпростіших діалогових програм.

***Завдання:*** скласти алгоритми у вигляді блок-схем для задач, які наведено нижче (обрати по три задачі [***за правилом,***](http://10.10.10.140/pr_trpz/z.txt)встановленим викладачем) та написати програми для їх розв’язання мовою програмування Сі. Виконати звіт до лабораторної роботи, який вміщує створені блок-схеми та програми.

**Завдання:**

|  |  |
| --- | --- |
| ***5*** | Напишіть програму для обчислення min{a,b,c}. |
| ***41*** | Написати програму, що підраховує символи пунктуації у рядку символів, що вводиться з клавіатури. |
| ***8*** | Дано три сторони трикутника A,B,C. Визначити його площу та перевірити, чи є він прямокутним. |

**Хід роботи**

**Завдання 8:**

1. Напишемо код мовою С++.

#include <iostream>

using namespace std;

double sq\_rec(double a, double b)

{

return 0.5 \* a \* b;

}

double sq\_ger(double a, double b, double c)

{

double p = (a + b +c) / 2;

return sqrt(p \* (p - a) \* (p - b) \* (p - c));

}

int main ()

{

system("color f0");

bool flag = true;

double a = 0, b = 0, c = 0;

//------------------------------Squere-----------------------

while(flag)

{

cout << "Please enter tree sides of the triangle" << endl;

cin >> a >> b >> c;

if(a + b > c && a + c > b && b + c > a)

{

flag = false;

cout << endl << "Squere of the triangle:" << sq\_ger(a, b, c) << endl;

if(sq\_ger(a, b, c) == sq\_rec(a, b))

cout << "This triangle is rectangular" << endl;

if(sq\_ger(a, b, c) == sq\_rec(b, c))

cout << "This triangle is rectangular" << endl;

if(sq\_ger(a, b, c) == sq\_rec(a, c))

cout << "This triangle is rectangular" << endl;

}

else cout << "Error of imput!!!" << endl << endl;

}

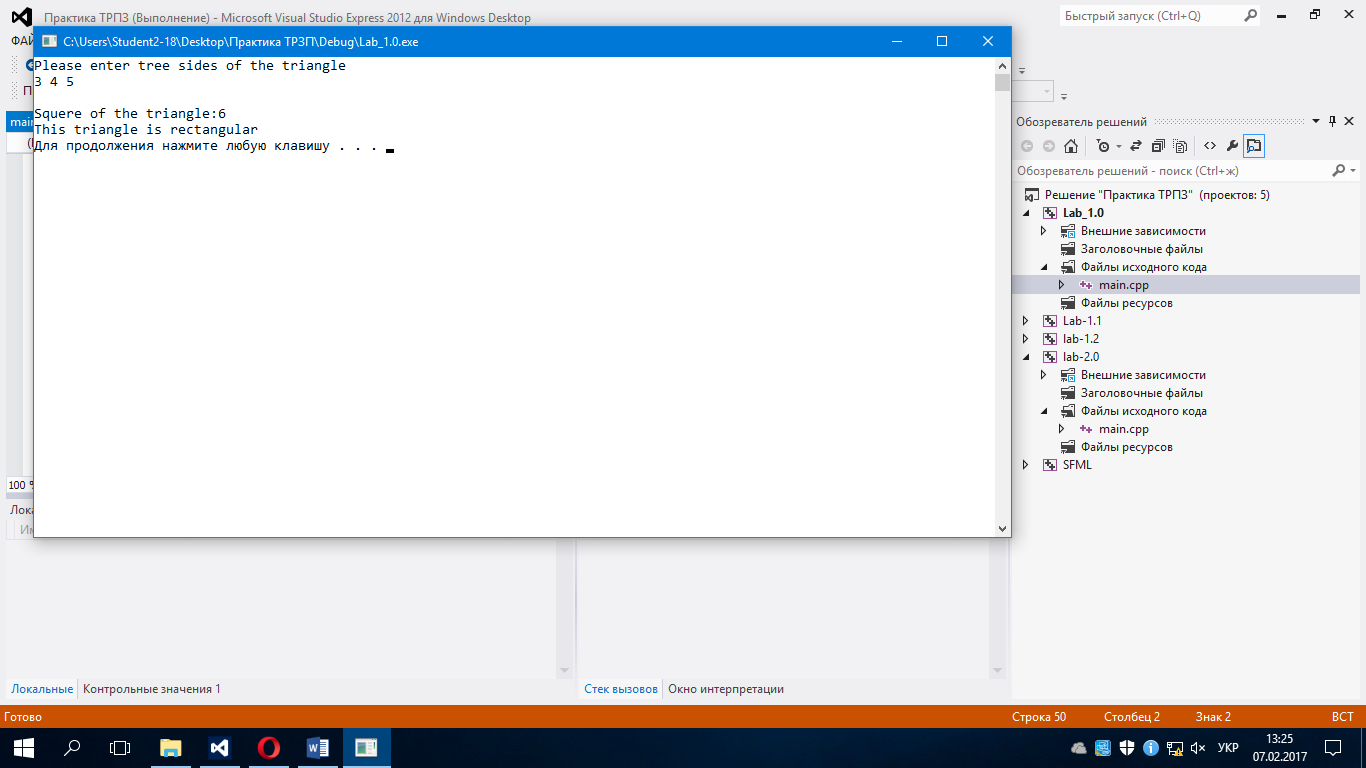
system("pause");

return 0;

}

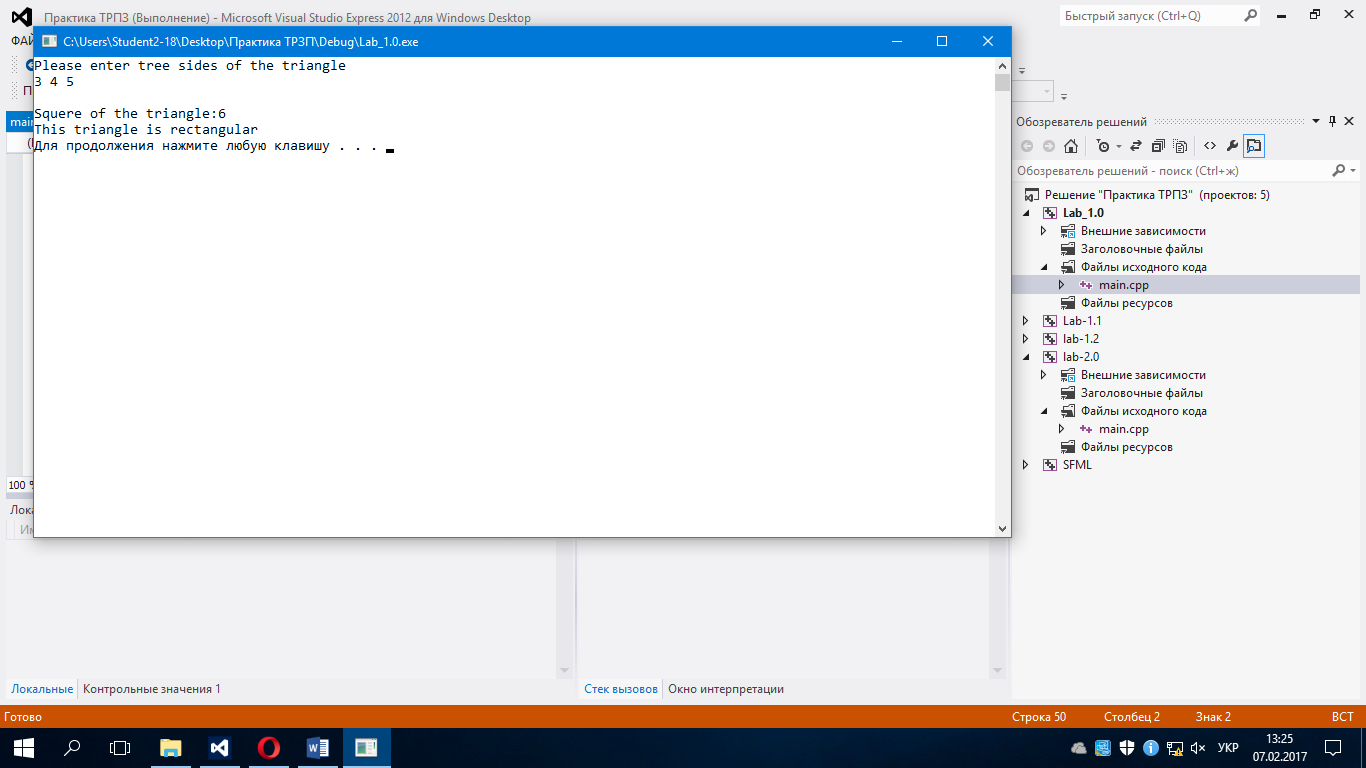
1. Тестування програми.

Якщо ми введемо такі сторони трикутника: 3, 4, 5. То програма маж вивести площу 6 та повідомити нас про те, що трикутник є прямокутним.



Результат підтвердився, програма працює вірно.

1. Результат виконання програми.



**Завдання 5:**

1. Напишемо програму мовою С++.

#include <iostream>

using namespace std;

int main()

{

system("color f0");

int a, b, c;

cout << "Enter the numbers(3):";

cin >> a >> b >> c;

int min = a;

if (b < min) {

min = b;

}

if (c < min) {

min = c;

}

cout << "Min: " << min << endl;

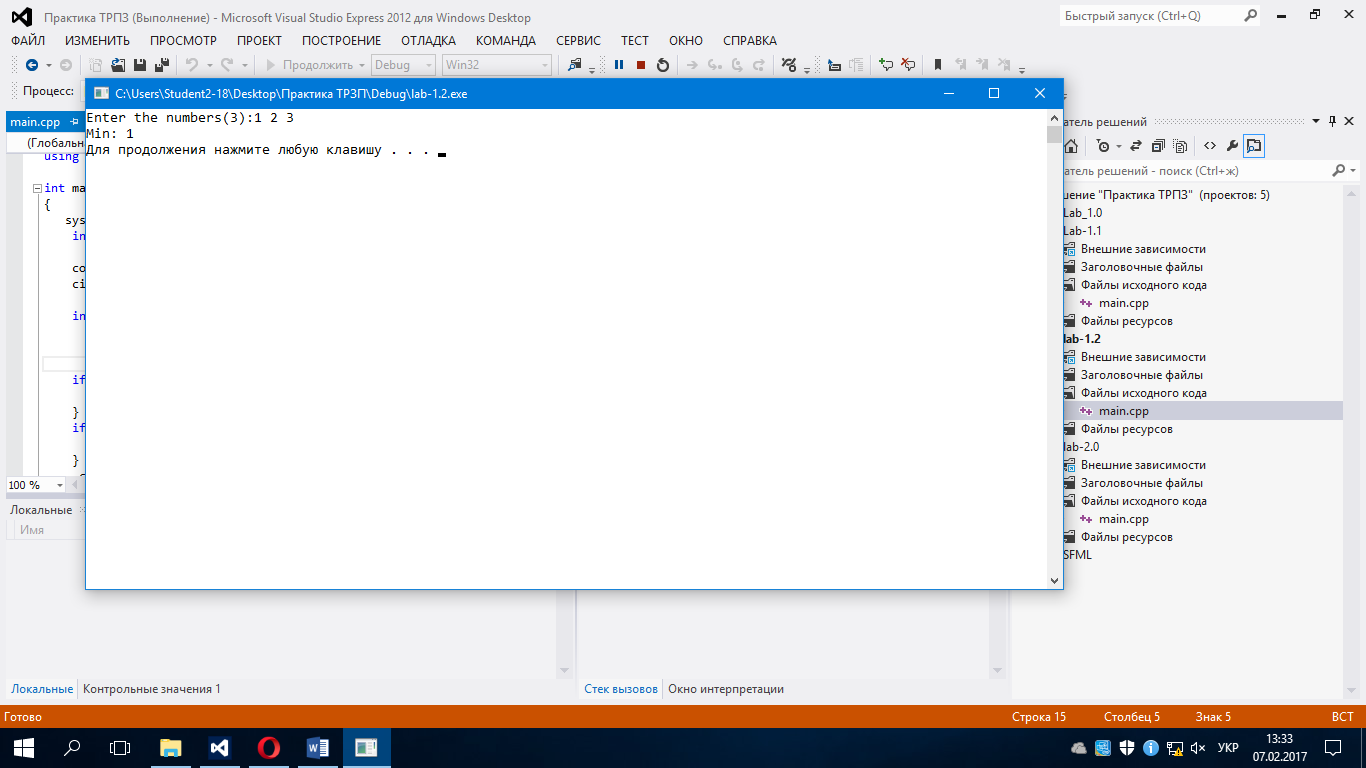
system("pause");

return 0;

}

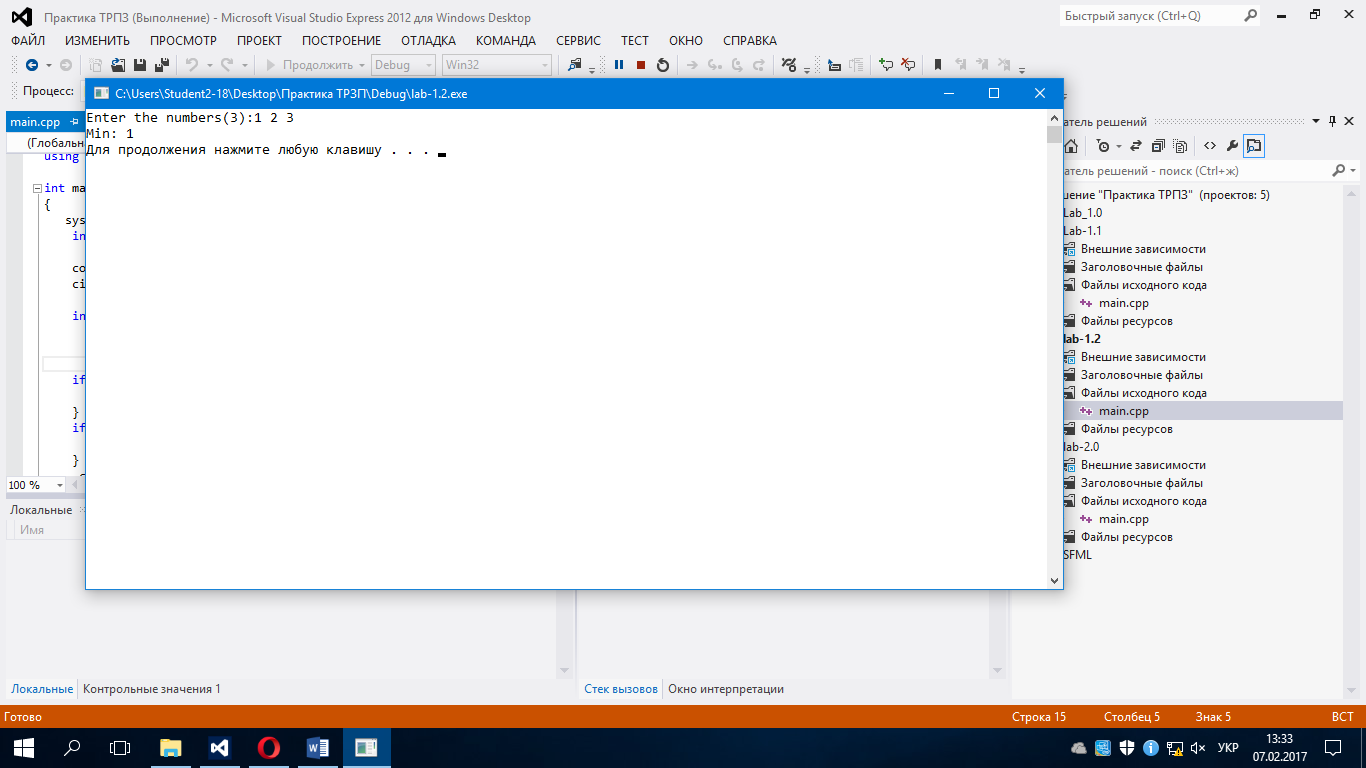
1. Тестування програми.

Ввівши три числа (1, 2, 3) програма має вивести мінімальне число на екран, тобто цифру 1.



Програма працює вірно.

1. Результат виконання програми.



**Завдання 41:**

1. Напишемо код мовою С++.

#include <iostream>

#include <string>

using namespace std;

bool ispunct(char str)

{

return str == '.' || str == ',' || str == '?' || str == '!' || str == '-' || str == ':' || str == ';';

}

int main ()

{

system("color f0");

int ix = 0, ctr = 0;

char str[1000];

cout << "Enter string, please!" << endl;

cin.get(str, 1000, '\n');

while (str[ix])

{

if ( ispunct(str[ix]) ) ctr++;

ix++;

}

cout << "The proposal contains " << ctr << " a punctuation character" << std::endl;

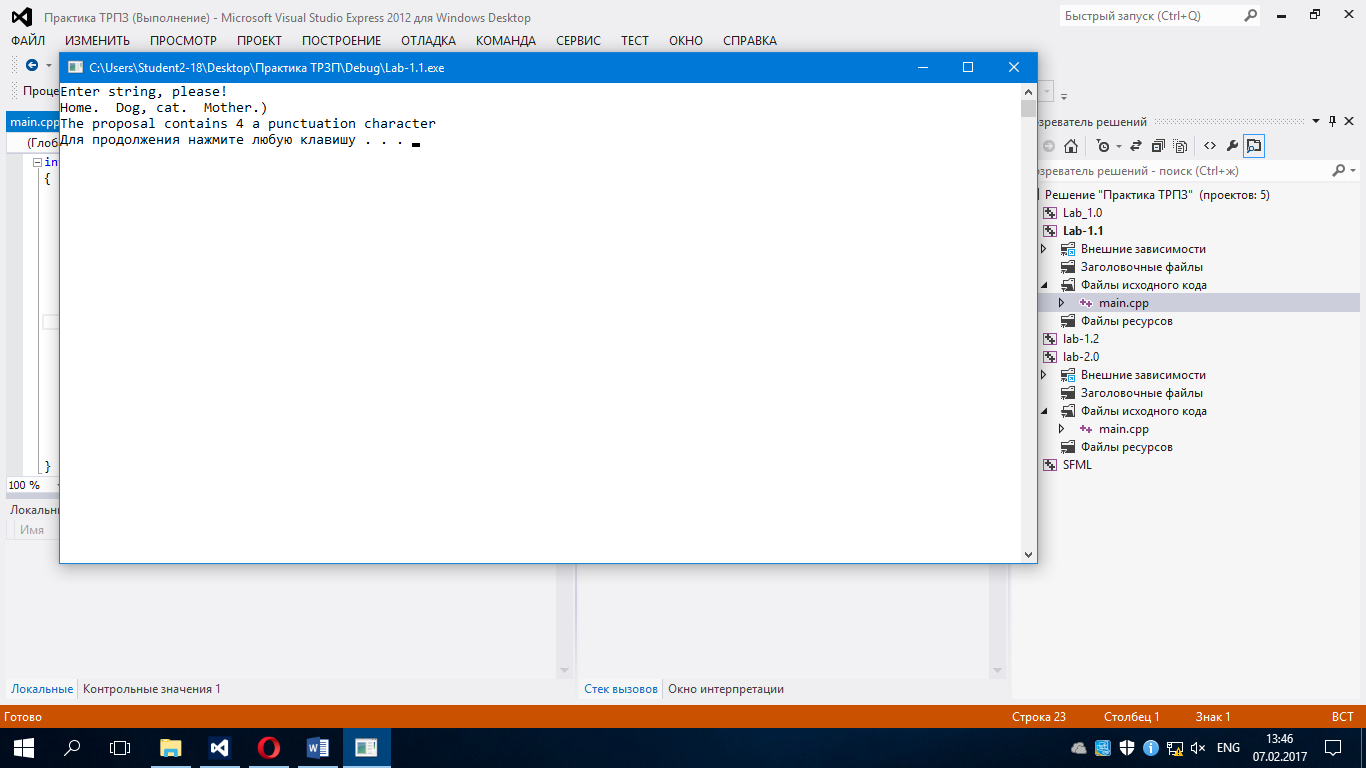
system("pause");

return 0;

}

1. Тестування програми.

Ввівши три рядок (Home. Dog, cat. Mother.) програма має вивести кількість знаків пунктуації в рядку, тобто цифру 4.



Програма працює правильно.

**Висновок:** Я ознайомився з простими типами даних. Базових конструкції мови С

## Лабораторна робота №2

**Тема:** цикли та розгалуження. Функції та їх застосування

**Мета роботи:** навчитися складати програми циклічних обчислювальних процесів, програми з використанням розгалуження та функцій користувача.

**Завдання:** вивести на екран у вигляді таблиці значення функції F на інтервалі від Xпоч до Xкін з кроком Н. Значення a, b, c, Xпоч, Хкін, Н – дійсні числа, вводяться з клавіатури. Задачу необхідно розв’язати з використанням функцій користувача. При застосуванні функцій використання глобальних змінних забороняється.

|  |  |  |
| --- | --- | --- |
| ***8*** | http://10.10.20.140/pr_trpz/lr2/Image8.gif | Функція *F* повинна приймати дійсне значення, якщо вираз (Ац МОД2 Вц) І НЕ(Ац ЧИ Сц) не дорівнює нулю, і ціле значення в протилежному випадку. Через Ац, Вц і Сц позначені цілі частини значень *а*, *b*, *c*, операції І, ЧИ і МОД2 (додавання за модулем 2) — порозрядні. |

Хід роботи

1. Напишемо код мовою С++.

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

void print\_f(double h, double rez)

{

cout << "|" << setw(5) << h << "|" << setw(5) << rez << "|" << endl;

}

void print\_int(double h, int rez)

{

cout << "|" << setw(5) << h << "|" << setw(5) << rez << "|" << endl;

}

bool isfunc(int a, int b, int c)

{

return ((a%2 == b) && !(a || b)) != 0;

}

int main ()

{

system("color f0");

double a = 0, b = 0, c = 0,

h = 0, xStart = 0, xFinish = 0, rez = 0;

cout << "Enter a, b, c." << endl;

cin >> a >> b >> c;

cout << "Enter Start, Finish, Step" << endl;

cin >> xStart >> xFinish >> h;

cout << "------------The table--------------------------"<< endl;

cout << "|" << setw(5) << "x" << "|" << setw(5) << "rez" << "|" << endl;

cout << "-----------------------------------------" << endl;

for(double x = xStart; x <= xFinish; x += h)

{

if(isfunc(a, b, c))

{

if(c < 0 && a != 0){

rez = (-a \* x \* x);}

else

if(c > 0 && a == 0){

rez = ((a - x)/(c \* x));}

else {rez = (x/c); }

}

else

{

if(c < 0 && a != 0)

rez = int(-a \* x \* x);

else

if(c > 0 && a == 0)

rez = int((a - x)/(c \* x));

else rez = int(x/c);

}

cout << "|" << setw(5) << x << "|" << setw(5) << rez << "|" << endl;

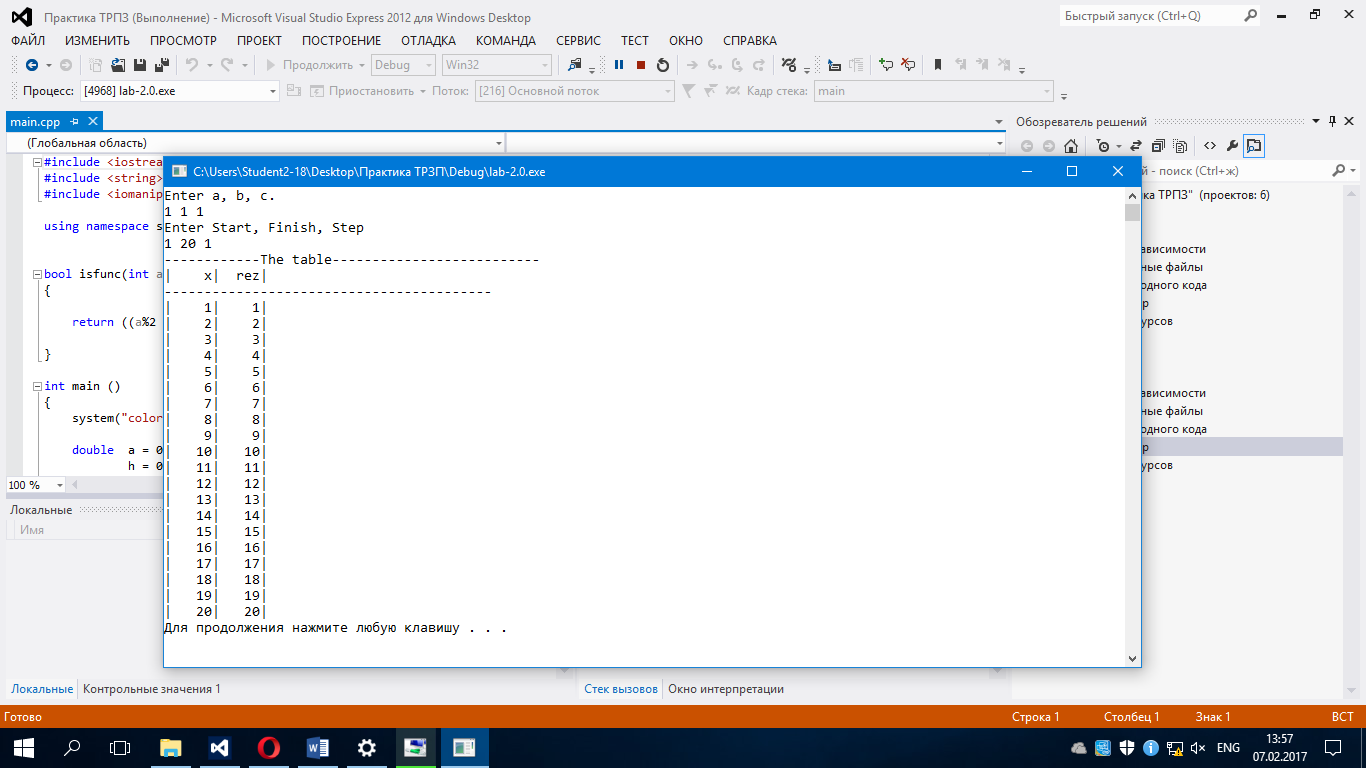
}

system("pause");

return 0;

}

1. Тестування програми.



**Висновок:** я навчився складати програми циклічних обчислювальних процесів, програми з використанням розгалуження та функцій користувача.

**Лабораторна робота №3**

**Тема:** одновимірні та багатовимірні масиви, робота з текстовими рядками

**Мета роботи:** навчитися складати програми обробки лінійних та двовимірних масивів, реалізовувати найпростіші операції з текстовими рядками.

**Завдання:** оформити кожен пункт завдання вибраного варіанту у вигляді функції. Всі необхідні дані для функцій передаються їм в якості параметрів. Використання глобальних змінних у функціях не допускається.

**Завдання:**

|  |  |  |
| --- | --- | --- |
| ***8*** | **1** | В одновимірному масиві, що складається з N дійсних елементів, обчислити:   номер мінімального елемента масиву;   суму елементів масиву, що розташовані між першим і другим від’ємними елементами.  Перетворити масив таким чином, щоб спочатку розташовувались всі елементи, модуль яких не перевищує 10, а потім – решта. |
| **2** | Характеристикою стовпця цілочисельної матриці назвемо суму модулів його від’ємних непарних елементів. Переставляючи стовпці заданої матриці, розташувати їх у відповідності із ростом характеристик. |
| **3** | З клавіатури вводиться текстовий рядок. Скласти програму, яка підраховує кількість цифр у тексті; виводить на екран слова, що починаються з приголосних літер; знищує всі слова, які починаються і закінчуються за одну й ту ж літеру. |

**Хід роботи**

**Звдання 1:**

1. **Напишемо код мовою С++**

#include <iostream>

#include <cmath>

using namespace std;

void temp(int \*Arr, int i)

{

int temp = Arr[i];

Arr[i] = Arr[i - 1];

Arr[i - 1] = temp;

}

void sort(int \*Arr, int N)

{

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

for(int j = 0; j < N; j++){

if(Arr[j] < 10 && Arr[j] > -10)

if(j == 0);

else

temp(Arr, j);

}

}

void sum(int \*Arr, int N, int \*Arr1)

{

int a = 0, b = 0, sum = 0;

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

if(Arr[i] < 0) {a = i; break;}

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

if(Arr[i] < 0 && i != a) {b = i; break;}

if(b == 0) { Arr1[0]= 0;}

else{

for(int i = a + 1; i < b; i++)

if(a+1 == b) {Arr1[1] = 0; Arr1[0]= 1; break;}

else

sum +=Arr[i];

}

Arr1[1]= sum;

}

void print(int \*Arr, int N)

{

cout << endl << "Elements array: ";

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

cout << Arr[i] << "\t";

}

void input(int \*Arr, int N)

{

cout << "Input element array:";

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

cin >> Arr[i];

}

int max(int Arr[], int N)

{

int i = 0;

int min = Arr[0], count = 0;

while ( i != N)

{

if(min > Arr[i])

{

min = Arr[i];

count = i;

}

i++;

}

return count+1;

}

int main ()

{

system("color f0");

int N = 5;

cout << "Input count array:";

cin >> N;

int \*Arr, \*Arr1;

Arr1 = new int[2];

Arr = new int[N];

input(Arr, N);

int n = max(Arr, N);

cout << "Min number:" << n << endl;

sum(Arr, N, Arr1);

if(Arr1[0] == 0)

cout << "Error" << endl;

else

cout << "Sum:" << Arr1[1] << endl;

sort(Arr, N);

print(Arr, N);

system("pause");

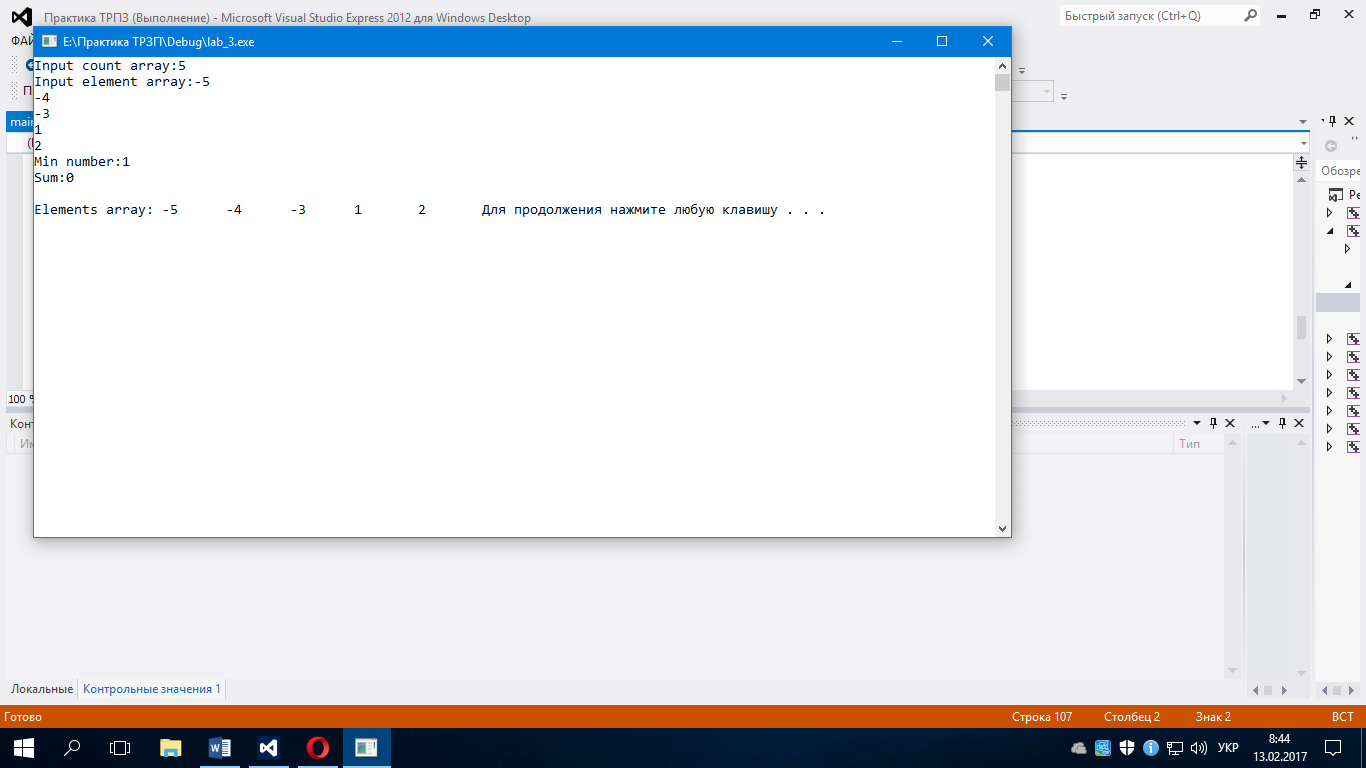
delete Arr;

delete Arr1;

return 0;

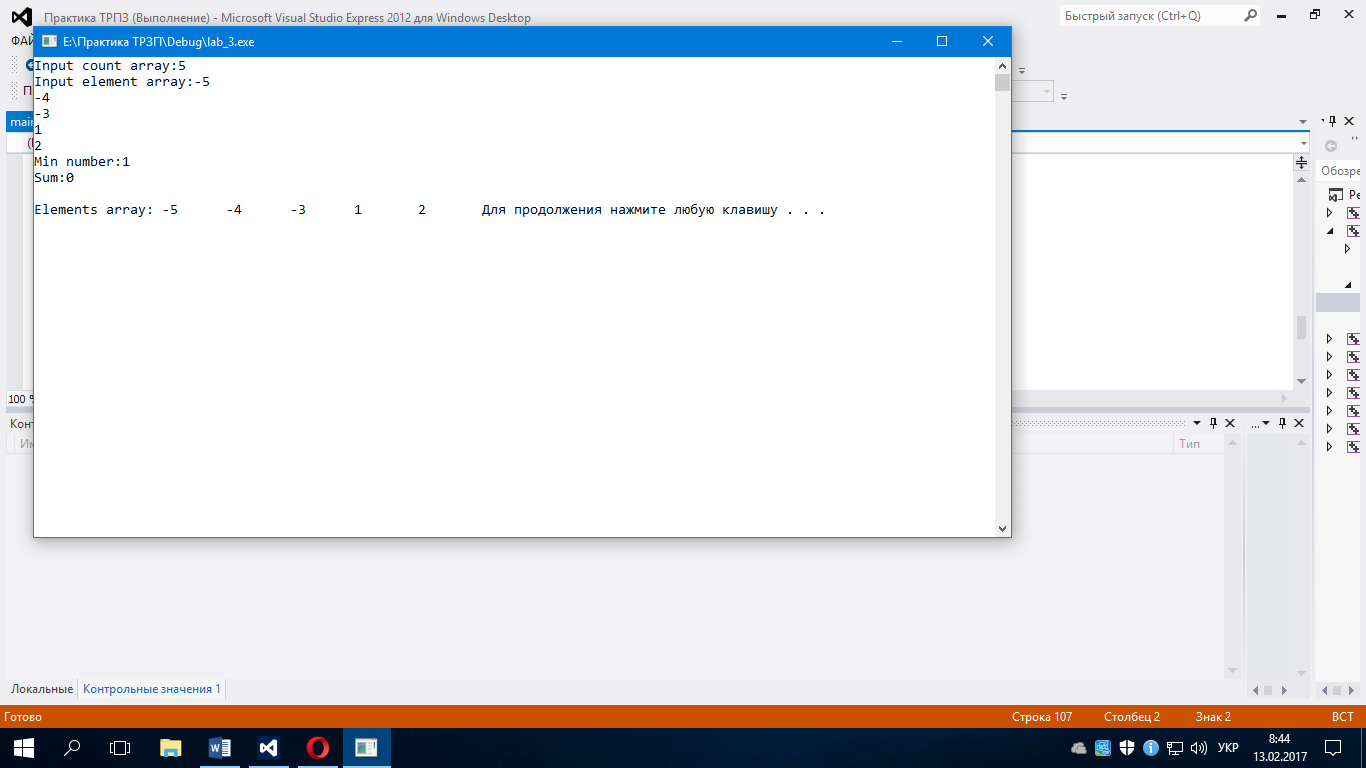
}

1. Тестування програми.



Програма працює вірно.

1. Результат вионання програми.



**Завдання 2:**

1. Напишемо код мовою С++.

#include <iostream>

using namespace std;

void Input(int \*\*A, int n, int m){

int i,j;

for(i=0;i<n;i++)

for(j=0;j<m;j++)

{

cout << "Vvedite a[" << i << "][" << j << "]: " ;

cin >>A[i][j];

}

}

void Output(int \*\*A, int n,int m){

int i,j;

for(i=0;i<n;i++){

printf("\n");

for(j=0;j<m;j++)

printf("\t%5d",A[i][j]);

}

}

int Sum(int \*\*A, int n, int j){

int i,S=0;

for(i=0;i<n;i++)

{

if(A[i][j]<0 && A[i][j]%2!=0)

{

S+=abs(A[i][j]);

}

}

return S;

}

void Sort(int \*\*A, int n, int m){

int i,l,j,S1,S2,a;

for(l=0;l<m;l++)

for(j=1;j<m;j++)

{

S1=Sum(A,n,j);

S2=Sum(A,n,j-1);

for(i=0;i<n;i++)

if(S1<S2)

{

a=A[i][j];

A[i][j]=A[i][j-1];

A[i][j-1]=a;

}

}

}

int main() {

system("color f0");

int \*\*A,i,j,n,m;

cout << "Input size matrix A[n][m]:" << endl;

cin >> n >> m;

A=(int\*\*)malloc(sizeof(int)\*n);

for(i=0; i<n; i++)

A[i]=(int\*)malloc(sizeof(int)\*m);

Input(A,n,m);

cout << endl << "Matrix:" << endl;

Output(A,n,m);

cout << endl << "Result:" << endl;

Sort(A,n,m);

Output(A,n,m);

for(i=0; i<n; i++)

free(A[i]);

free(A);

cout << endl;

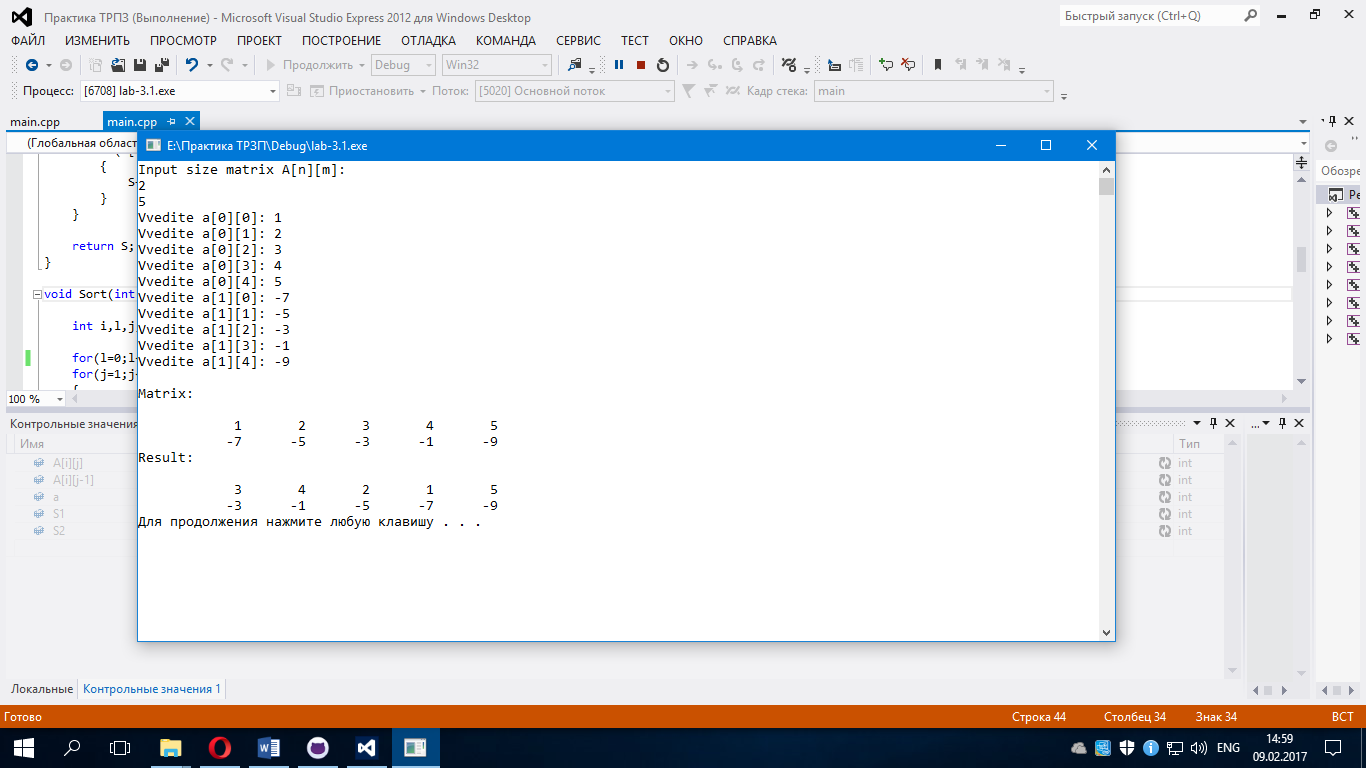
system("pause");

return 0;

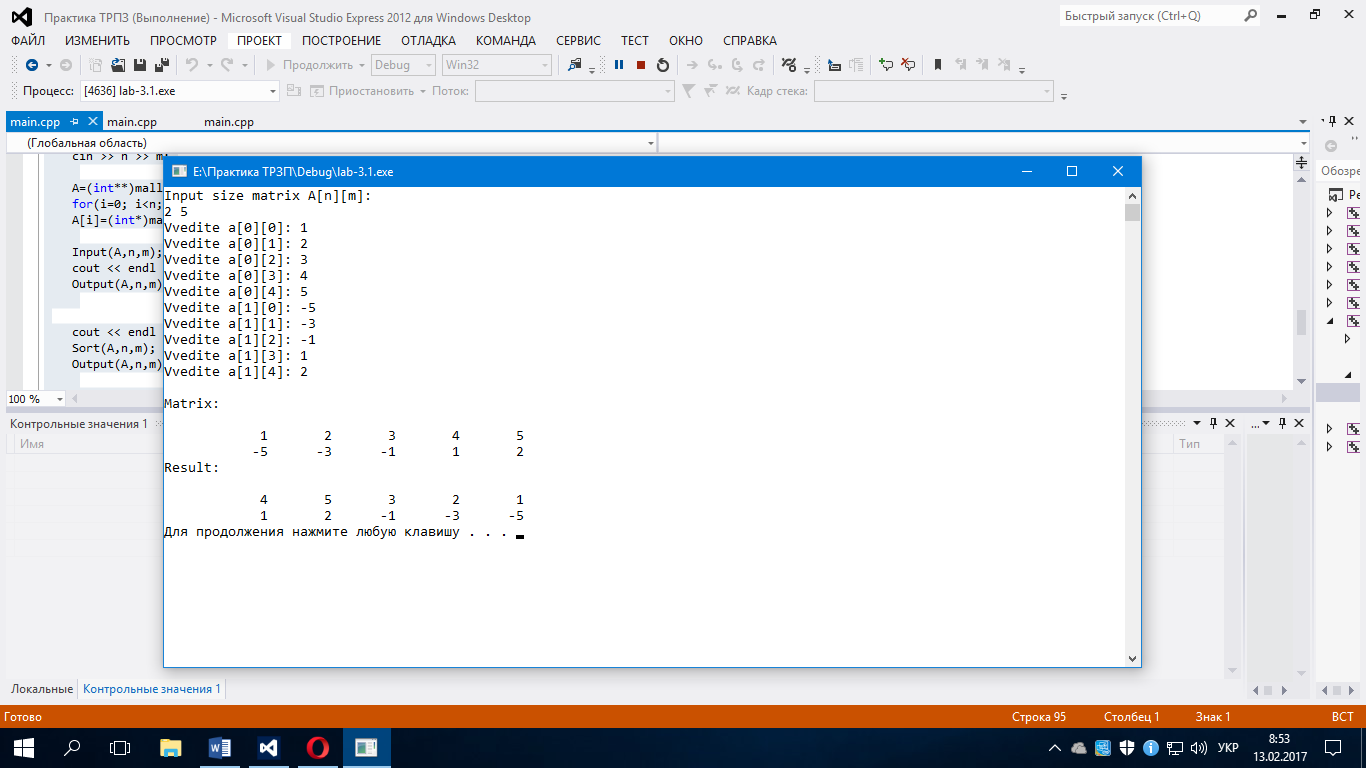
}

1. Тестування програми.

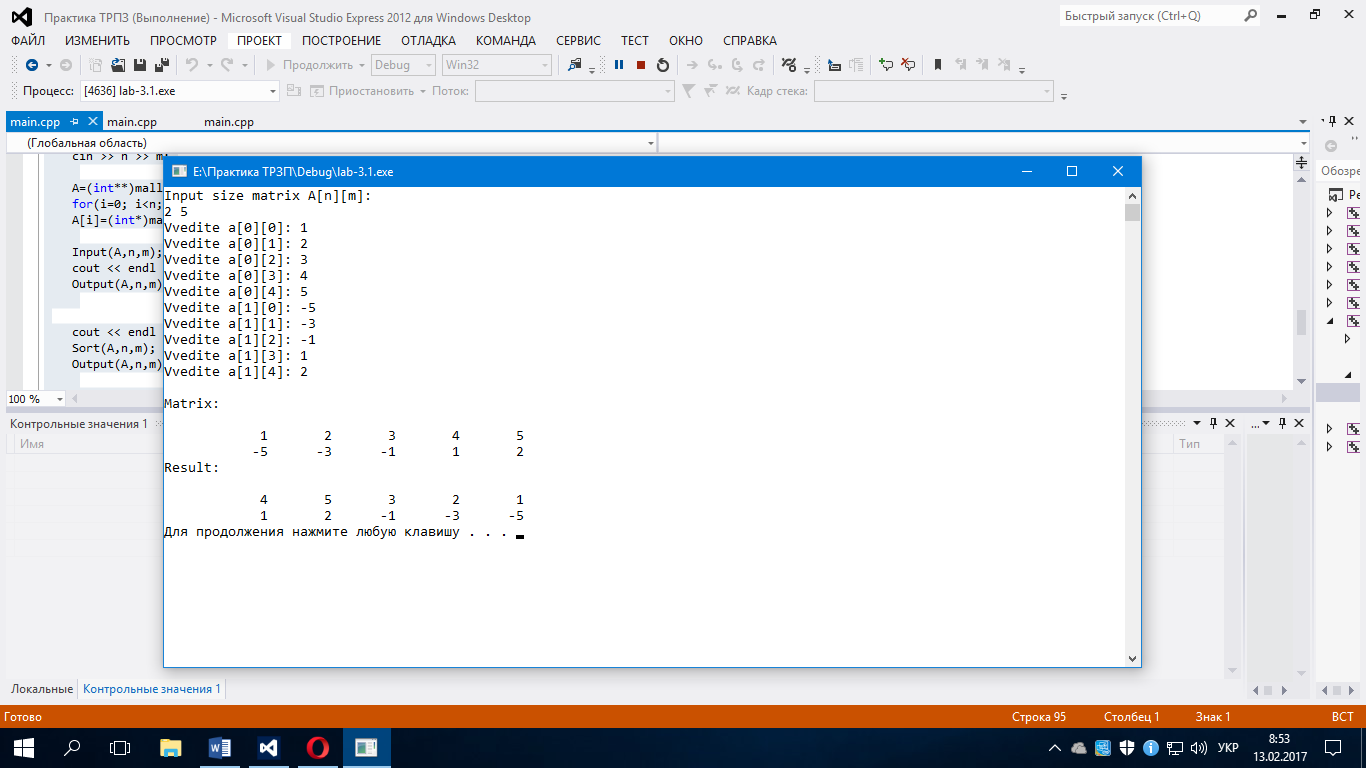
При написанні програми я допустив помилку, в результаті чого вона працювала не вірно. Вконуючи програму покроково, мені вдалося виправити її. Проблема була в завнішньому циклі сортування, який проходив недостатню кількість разів.



Програм працює вірно, вісі помилки виправлені.



1. Результат виконання програми.



Завдання 3:

1. Напишемо код мовою С++.

#include <iostream>

#include <string>

using namespace std;

void erase( char \* str)

{

for(int i = 0; i < strlen(str); i++)

{

if(str[i] == ' ' || str[i] == '!' || str[i] == '?' || str[i] == ',' || str[i] == '.' || str[i] == ':' || str[i] == ';') continue;

for(int j = i; j < strlen(str); j++)

{

if(str[j] == ' ' || str[j] == '!' || str[j] == '?' || str[j] == ',' || str[j] == '.' || str[j] == ':' || str[j] == ';') continue;

if(str[j + 1] == ' ' || str[j + 1] == '\0' || str[j + 1] == '!' || str[j + 1] == '?' || str[j + 1] == ',' || str[j + 1] == '.' || str[j + 1] == ':' || str[j + 1] == ';')

{

if(str[i] == str[j])

{

for(int l = i; l < strlen(str); l++)

{

str[l] = str[l + (j - i) + 1];

}

//cout << endl << "TEST string:" << str << endl;

i = j = 0;

break;

}

else

{

i = j + 1;

break;}}}}}

bool is\_it(char\* str)

{

return str[0] == 'q' || str[0] =='Q' || str[0] == 'w' || str[0] == 'W' || str[0] == 'r' || str[0] == 'R' || str[0] == 't' || str[0] == 'T' || str[0] == 'p' || str[0] == 'P' || str[0] == 's' || str[0] == 'S' || str[0] == 'd' || str[0] == 'D' || str[0] == 'f' || str[0] == 'F' || str[0] == 'g' || str[0] == 'G' || str[0] == 'h' || str[0] == 'H' || str[0] == 'l' || str[0] == 'L' || str[0] == 'k' || str[0] == 'K' || str[0] == 'z' || str[0] == 'Z' || str[0] == 'x' || str[0] == 'X' || str[0] == 'c' || str[0] == 'C' || str[0] == 'v' || str[0] == 'V' || str[0] == 'b' || str[0] == 'B' || str[0] == 'n' || str[0] == 'N' || str[0] == 'm' || str[0] == 'M';

}

int NumOfDigits(char\* str)

{

int count = 0;

cout << endl << "Len string:" << strlen(str)<< endl;

for(int i = 0; i < strlen(str); i++)

{

if(isdigit(str[i]))

count++;

}

return count;

}

int main()

{

char numbers[999];

char numbers1[999];

cout << "Input string: ";

cin.get(numbers, 1000, '\n');

for( int i = 0; i < 999; i++)

numbers1[i] = numbers[i];

cout << endl << "New string:" << numbers1 << endl;

erase(numbers1);

cout << endl << "Erases string:" << numbers1 << endl;

cout << "Numbers in string: " << NumOfDigits(numbers) << endl;

cout << "Word: ";

for(char \*p = strtok(numbers, " "); p; p = strtok(0, " "))

{

if(is\_it(p))

{

cout << p << " ";

}

}

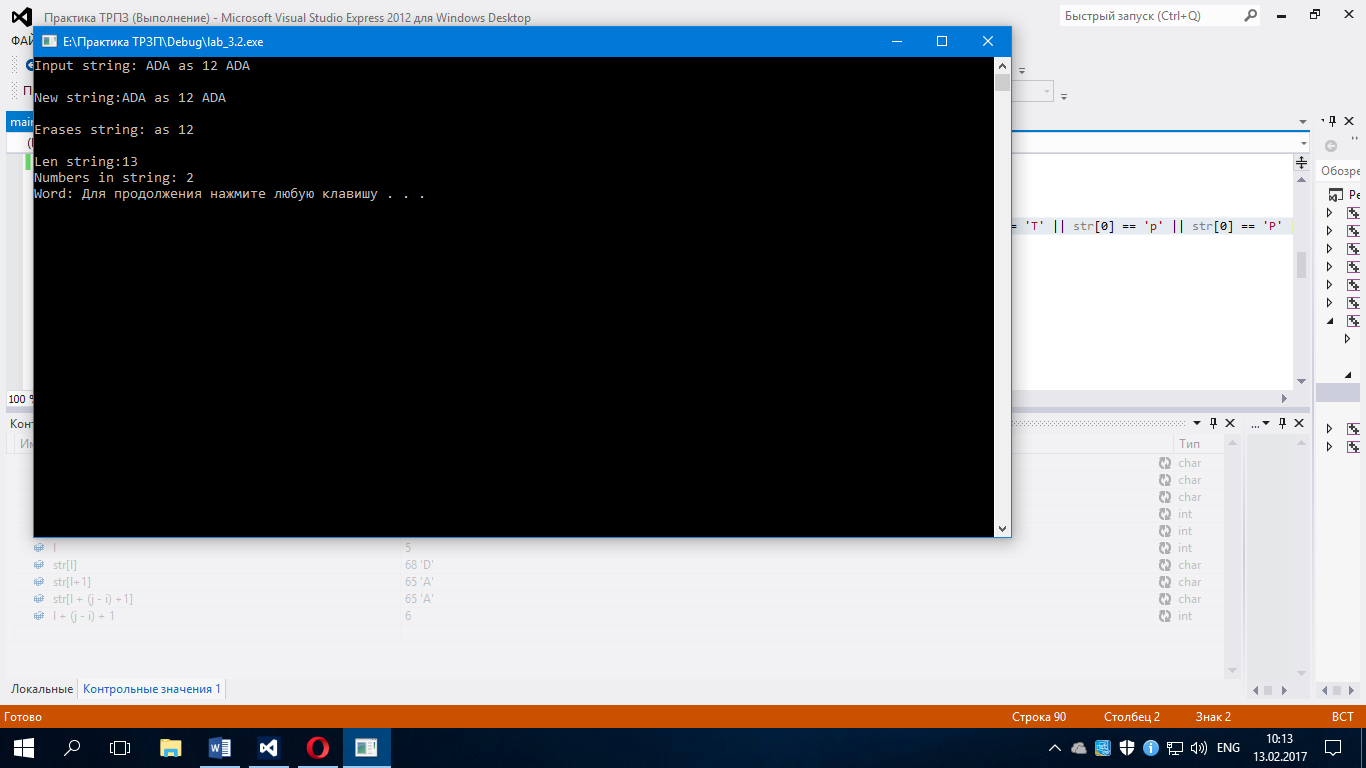
system("pause");

return 0;

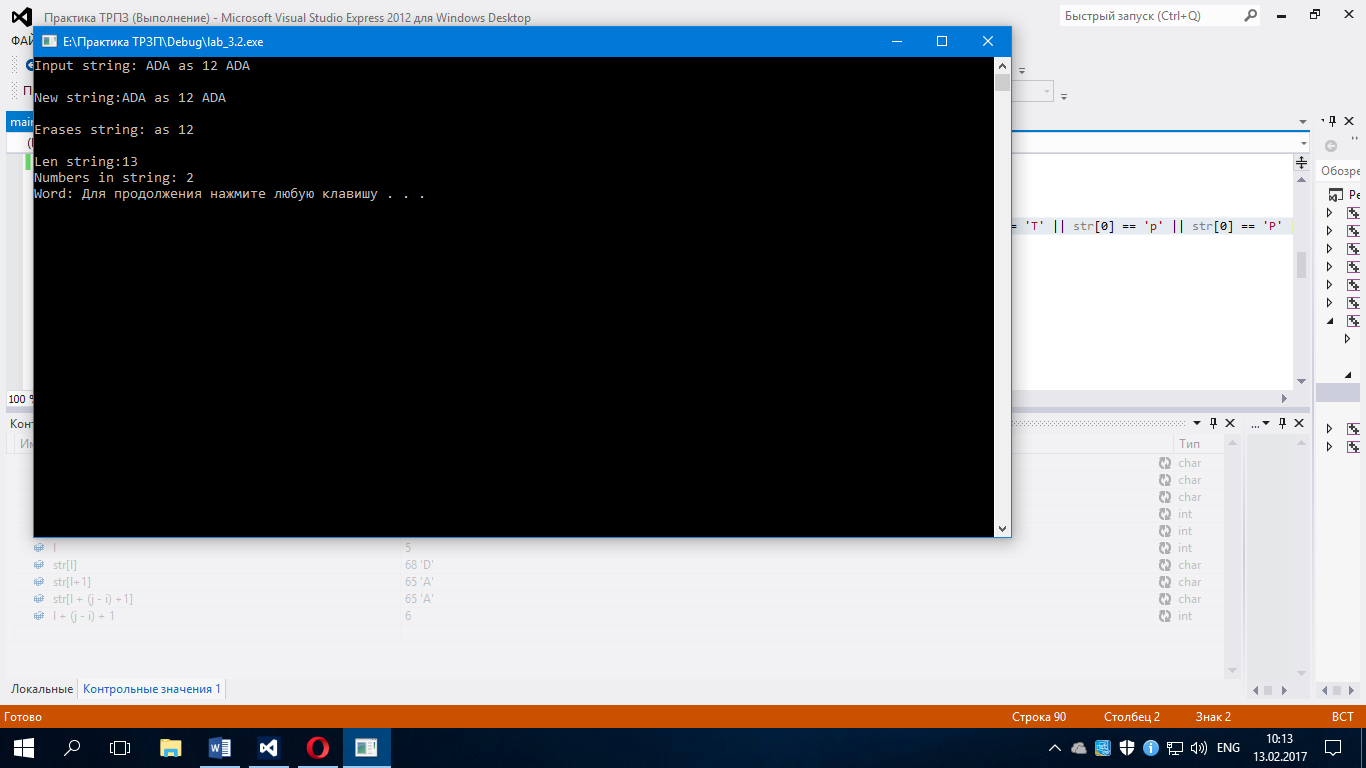
}

1. Тестування програми.

Програма працює вірно.



1. Результат виконання прогами.



**Висновок:** я навчився складати програми обробки лінійних та двовимірних масивів, реалізовувати найпростіші операції з текстовими рядками.

**Лабораторна робота №4**

**Тема:** структури та їх використання. Масиви структур. Використання динамічної пам’яті

**Мета роботи**: володіти практичними навичками використання структур та масивів структур, навчитися складати програми для виконання операцій з полями структур, навчитися використовувати динамічне виділення пам’яті.

**Завдання:** оформити кожен пункт завдання вибраного варіанту у вигляді функції. Всі необхідні дані для функцій передаються їм в якості параметрів. Використання глобальних змінних у функціях не допускається.

**Завдання:**

|  |  |  |
| --- | --- | --- |
| ***8*** | **1** | Описати структуру з ім’ям TIMETABLE, яка містить наступні поля: NAZV – назва пункту призначення; NUMR – номер поїзда; DATE – дата відправлення; TIME – час відправлення.  Написати програму, що окремими функціями виконує наступні дії: введення з клавіатури даних в масив TRAIN, що складається з N змінних типу TIMETABLE;  впорядкування записів за датою та часом відправлення поїзда; виведення на екран інформації про поїзди, що направляються в пункт призначення, назва якого введена з клавіатури; якщо таких поїздів немає, то вивести відповідне повідомлення. |
| **2** | Виконати завдання №2 з попередньої лабораторної роботи №3 використовуючи динамічне виділення пам'яті (варіант 13). |

**Хід роботи**

**Завдання 1:**

1. Напишемо код мовою С++.

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

struct DATE

{

int day;

int month;

int year;

}

struct TIME

{

int minute;

int hour;

};

struct TIMETABLE

{

string name;

int number ;

DATE date;

TIME time;

};

void Input(TIMETABLE\*& A, const int n)

{

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

{

cout << "Enter name: ";

getline(cin, A[i].name);

cout << "Enter number: ";

cin >> A[i].number;

cout << "Enter date of start: " << endl;

cout << "\t"<< "Enter day, month, year:";

cin >> A[i].date.day >> A[i].date.month >> A[i].date.year;

cout << "Enter time of start: " << endl;

cout << "\t"<< "Enter hour, minute:";

cin >> A[i].time.hour >> A[i].time.minute;

cin.ignore();

cin.clear();

cout << "------------------------------------------------------------------" << endl;

}

}

void Output(const TIMETABLE\* A, const int n)

{

cout << setw(16) << "Name" << setw(10) << "Number" << setw(13) << "DATE" << setw(15) << "TIME" << endl;

cout << "------------------------------------------------------------------" << endl;

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

{

cout << setw(16) << A[i].name << setw(8) << A[i].number << setw(10) << A[i].date.day << "/"<< A[i].date.month << "/"<< A[i].date.year << setw(10) << A[i].time.hour << ":"<< A[i].time.minute ;

cout << endl;

}

}

void Output\_par(const TIMETABLE\* TRAIN, const int N)

{

string name;

int flag = 0;

cout << "Input name:";

cin >> name;

cout << endl;

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

{

if(TRAIN[i].name == name)

{

cout << setw(16) << "Name" << setw(10) << "Number" << setw(13) << "DATE" << setw(15) << "TIME" << endl;

cout << "------------------------------------------------------------------" << endl;

cout << setw(16) << TRAIN[i].name << setw(8) << TRAIN[i].number << setw(10) << TRAIN[i].date.day << "/"<< TRAIN[i].date.month << "/"<< TRAIN[i].date.year << setw(10) << TRAIN[i].time.hour << ":"<< TRAIN[i].time.minute ;

cout << endl;

}

}

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

{

if(TRAIN[i].name == name) flag++;

}

if (!flag) cout << "Not trains!!" << endl;

}

void Sort(TIMETABLE \*& A, const int n)

{

TIMETABLE temp;

for (int i(0); i < n - 1; i++)

{

if(A[i].date.year > A[i+1].date.year)

{

temp = A[i];

A[i] = A[i + 1];

A[i + 1] = temp;

}

if(A[i].date.year == A[i+1].date.year)

if(A[i].date.month > A[i+1].date.month)

{

temp = A[i];

A[i] = A[i + 1];

A[i + 1] = temp;

}

if(A[i].date.year == A[i+1].date.year)

if(A[i].date.month == A[i+1].date.month)

if(A[i].date.day > A[i+1].date.day)

{

temp = A[i];

A[i] = A[i + 1];

A[i + 1] = temp;

}

if(A[i].date.year == A[i+1].date.year)

if(A[i].date.month == A[i+1].date.month)

if(A[i].date.day == A[i+1].date.day)

if(A[i].time.hour > A[i+1].time.hour)

{

temp = A[i];

A[i] = A[i + 1];

A[i + 1] = temp;

}

if(A[i].date.year == A[i+1].date.year)

if(A[i].date.month == A[i+1].date.month)

if(A[i].date.day == A[i+1].date.day)

if(A[i].time.hour == A[i+1].time.hour)

if(A[i].time.minute > A[i+1].time.minute)

{

temp = A[i];

A[i] = A[i + 1];

A[i + 1] = temp;

}

}

}

int main()

{

const int N = 2;

TIMETABLE\* TRAIN = new TIMETABLE[N];

Input(TRAIN, N);

Sort(TRAIN, N);

Output(TRAIN, N);

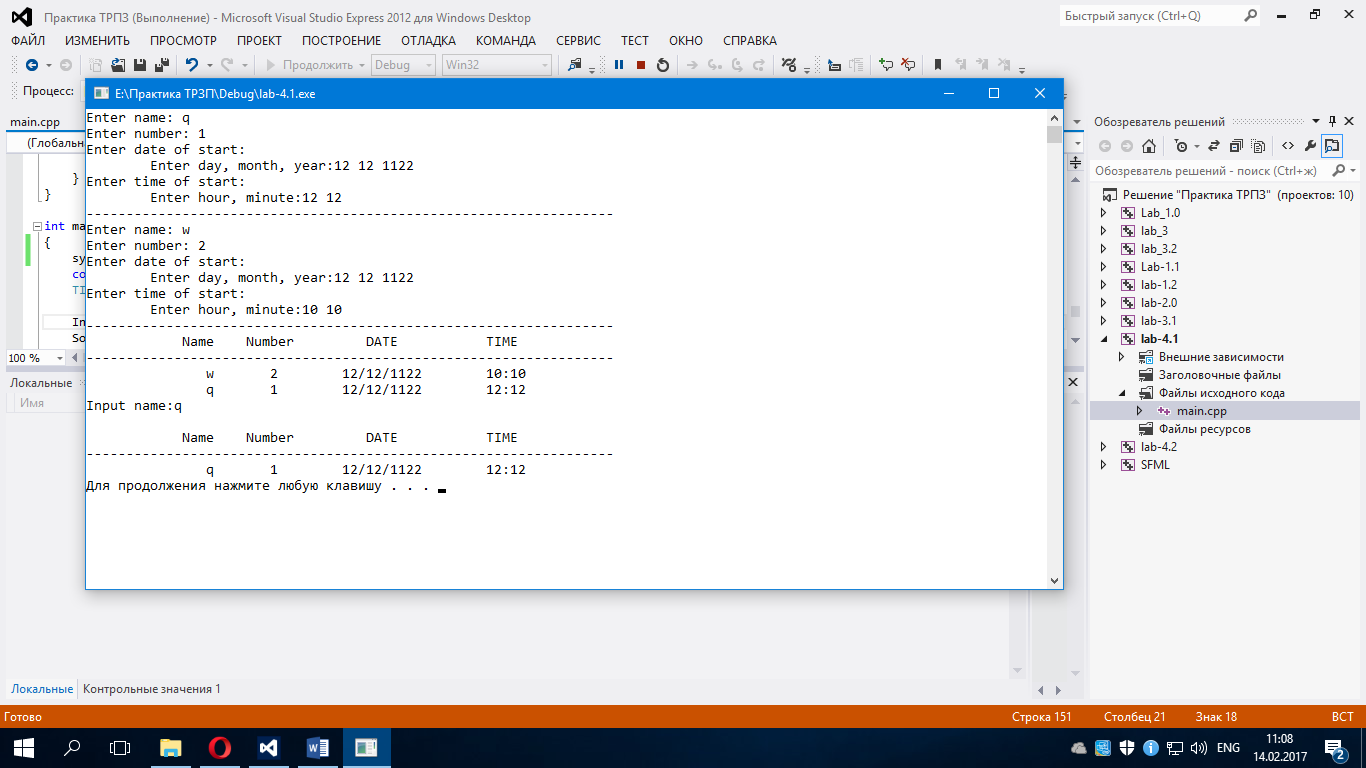
Output\_par(TRAIN, N);

system("pause");

return 0;

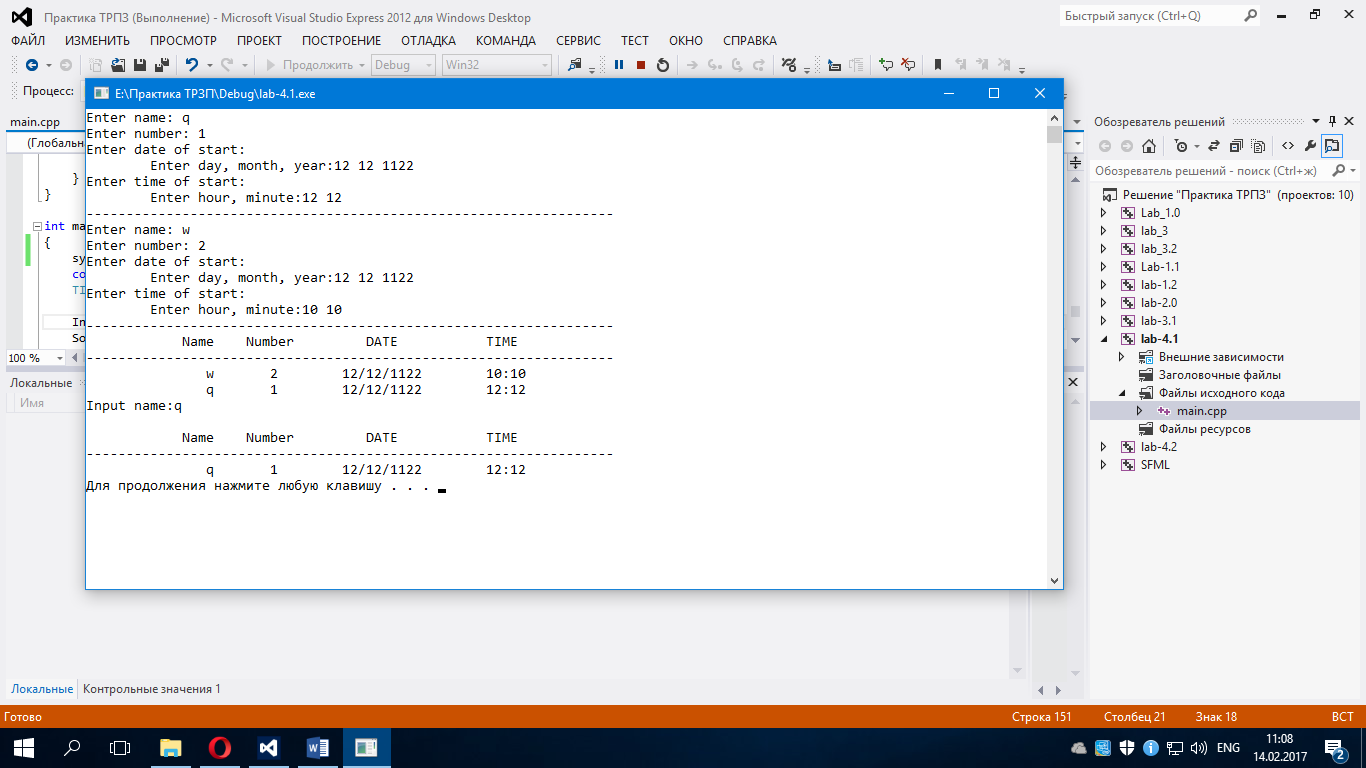
}

1. Тестування програми.



Програма працює вірно.

1. Результат виконання програми.



**Завдання 2:**

1. Напишемо код мовою С++.

#include <iostream>

using namespace std;

void Input(int \*\*A, int n){

int i,j;

for(i=0;i<n;i++)

for(j=0;j<n;j++)

{

cout << "Vvedite a[" << i << "][" << j << "]: " ;

cin >>A[i][j];

}

}

void Output(int \*\*A, int n){

int i,j;

for(i=0;i<n;i++){

printf("\n");

for(j=0;j<n;j++)

printf("\t%5d",A[i][j]);

}

}

int Sum(int \*\*A, int n){

int i,S=0;

for(i = 0; i < n; i++)

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

if(A[i][j] == 0)

S++;

return S;

}

int Max(int \*A, int n){

int max = A[0], count = 0;

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

if(A[i] > max)

{

max = A[i];

count = i;

}

return count;

}

void Sort(int \*\*A, int \*B, int n){

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

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

for(int l = 0; l < n; l++)

if(A[j][i] == A[l][i])

B[i]++;

}

int main() {

system("color f0");

int \*\*A,i,j,n, \*B;

cout << "Input size matrix(A[n][n]):" << endl;

cin >> n;

A=(int\*\*)malloc(sizeof(int)\*n);

for(i=0; i<n; i++)

A[i]=(int\*)malloc(sizeof(int)\*n);

B = (int\*)malloc(sizeof(int) \* n);

Input(A,n);

cout << endl << "Matrix:" << endl;

Output(A,n);

Sort(A, B, n);

cout << endl << "Count '0':" << Sum(A, n) << endl << "Number:" << Max(B,n) << endl;

for(i=0; i<n; i++)

free(A[i]);

free(A);

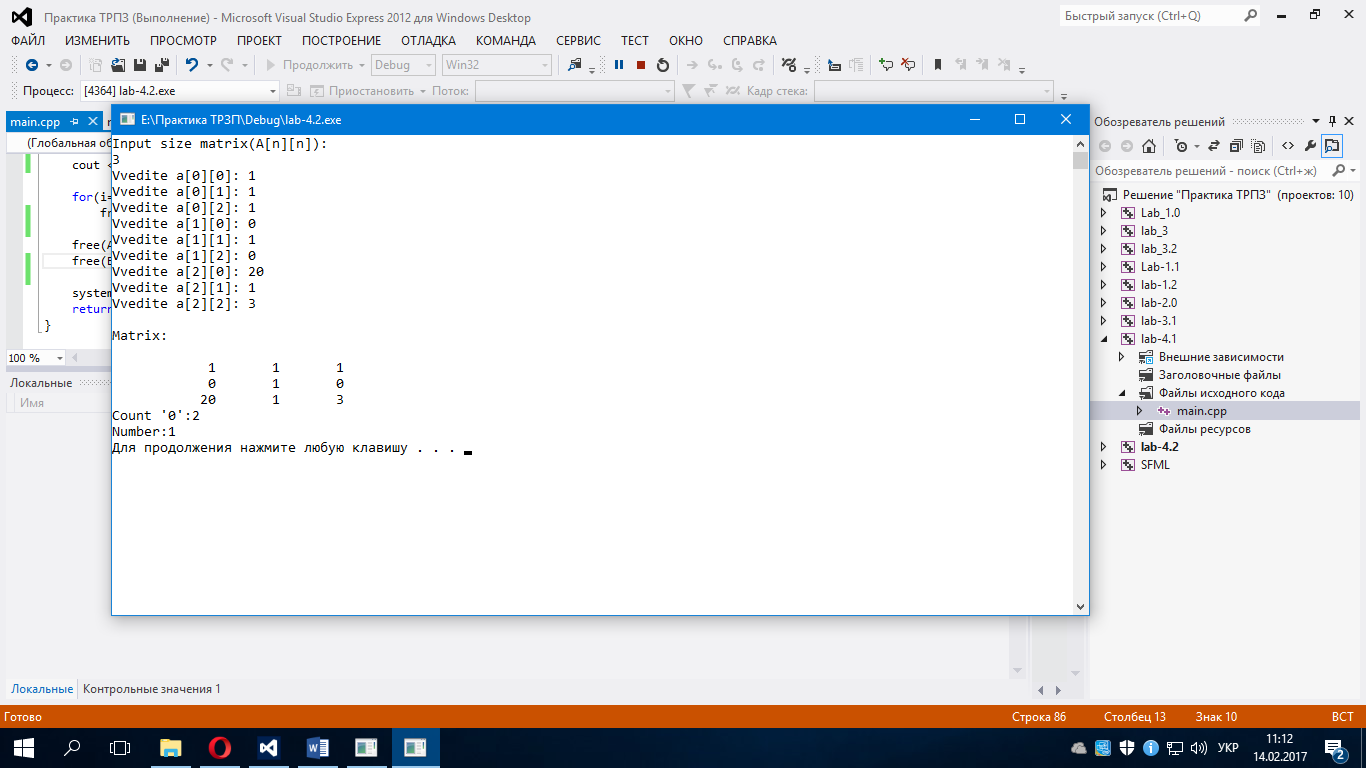
free(B);

system("pause");

return 0;

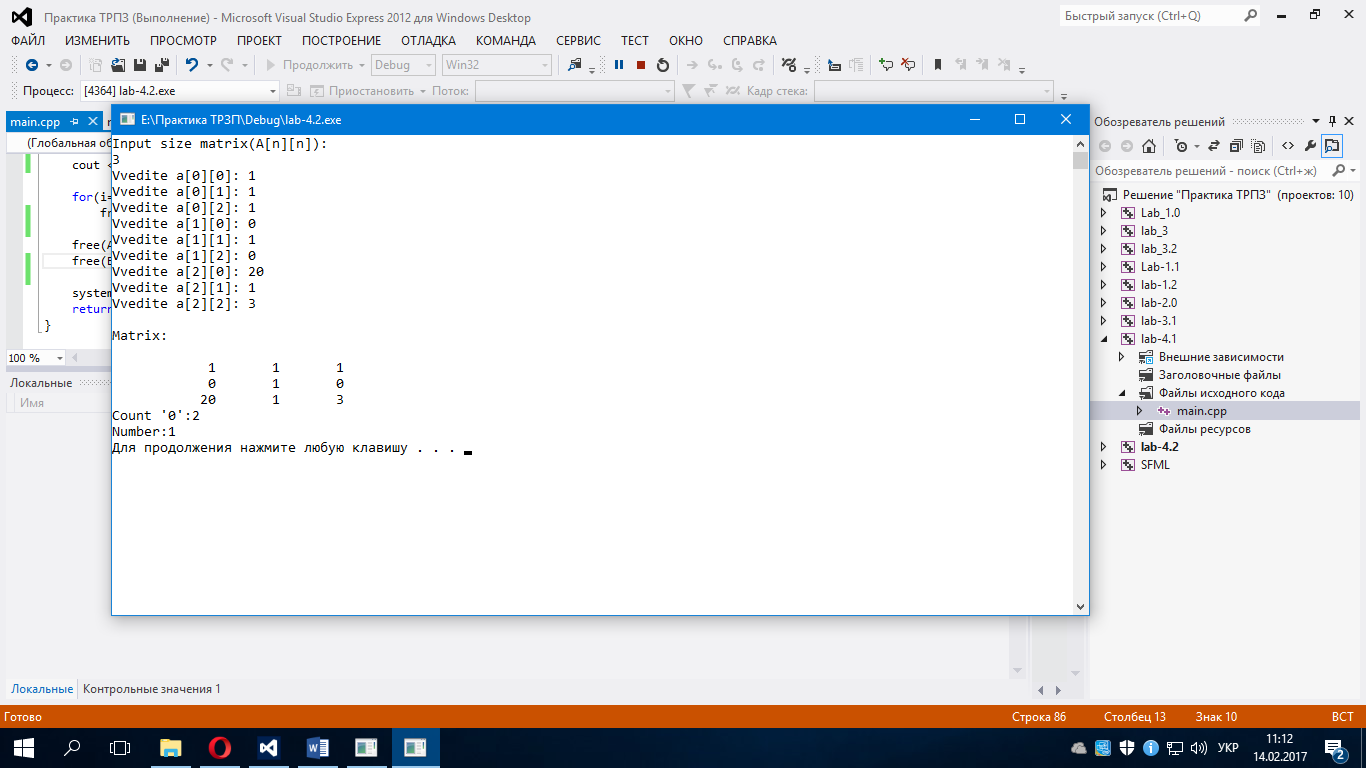
}

1. Тестування прогами.



Програма працює вірно.

1. Результат виконання програми.



**Висновок:** я навчився володіти практичними навичками використання структур та масивів структур, навчився складати програми для виконання операцій з полями структур, навчився використовувати динамічне виділення пам’яті.

Лабораторна робота №5

**Лабораторна робота №6**

**Тема:** ознайомлення з середовищем розробки Microsoft Visual C++ 6.0.

**Мета роботи: з**добути навики використання середовища розробки Microsoft Visual C++ 6.0, створити проект, що містить консольну програму, ознайомитися зі структурою проекту Visual C++, структурою консольної програми, здобути  
навички написання, компіляції, виправлення помилок та відлагоджування  
програм.

**Хід роботи**

1. Напишемо код мовою С++.

#include <iostream>

#include <cmath>

using namespace std;

int main(int argc, char\*\* argv)

{

const int a = 10;

int b = 4;

const float Pi = 3.13159;

const char\* ch = "Hello World!!!";

bool sw = true;

cout << "a = " << a << ",\t b = " << b << endl;

cout << "Rezult (b+a): " << b + a << endl; //14

cout << "Rezult (b-a): " << b - a << endl; //-6

cout << "Rezult (b / Pi): " << b / Pi << endl; //1.27324

cout << "Rezult (b \* Pi): " << b \* Pi << endl; //12.56636

cout << "Rezult (b % 2): " << b % 2 << endl; //0

cout << "Rezult (b << 1): " << (b << 1) << endl; //8 (bin = 1000)

cout << "Rezult (b >> 2): " << (b >> 2) << endl; //1 (bin = 1)

cout << "Rezult (!((b>a)&&((a>1)||(a<1)))): " << !((b>a)&&((a>1)||(a<1))) << endl; //1

cout << "Rezult (b > a): " << ((b > a) ? b : a) << endl; //10

cout << "Rezult (b += a): " << (b += a) << endl; //14

cout << "Rezult (b -= a): " << (b -= a) << endl; //4

cout << "Rezult (b &= a): " << (b &= a) << endl; //0 (0100 & 1010 = 0) побітове множення

cout << "Rezult (b |= a): " << (b |= a) << endl; //10 (0 | 1010 = 1010) побітове додавання

cout << "Rezult (b ^= a): " << (b ^= a) << endl; //0 (1010 ^ 1010 = 0) взаємне виключення

cout << "Rezult (sizeof(b)): " << sizeof(b) << endl; //4 byte

int y = 0, sum = 0;

for (int x(-20); x <= 20; x++)

{

y = pow(x, 5) - 5\*pow(x, 3) + x + 37;

if ((y % 13) == 5)

sum += y;

}

cout << endl << "Sum = " << sum << endl;

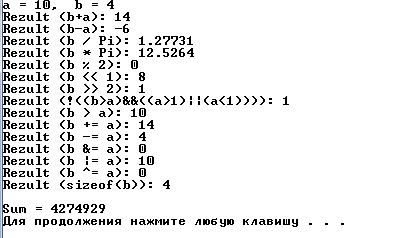
system("pause");

return NULL;

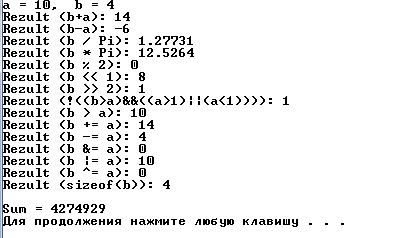
}

1. Тестування програми.

Програма працює вірно.



1. Результат виконання програми.



**Висновок:** я **з**добути навики використання середовища розробки Microsoft Visual C++ 6.0, створити проект, що містить консольну програму, ознайомитися зі структурою проекту Visual C++, структурою консольної програми, здобути  
навички написання, компіляції, виправлення помилок та відлагоджування  
програм.

**Лабораторна робота №7**

**Тема:** функції користувача та передача параметрів в функцію.  
**Мета роботи:** отримати практичні навики роботи з функціями користувача, навчитися передавати аргументи в функцію, створювати та викликати власні функції, використовувати засоби умовної компіляції.

**Хід роботи**

1. Напишемо код мовою С++.

#include <iostream>

#include <cmath>

#define ROVNO =

#define PLUS +

#define MINUS -

#define DEFINED 1

#define DILENNJA /

#define OSTACHA %

#define ZSUV\_RIGHT >>

using namespace std;

void Sum\_if()

{

int y = 0, sum = 0;

for (int x(-10); x <= 20; x++)

{

y = 00.1\*exp(-0.5\*x)+0.05\*pow(x,3);

if (y < 0 || y > 90 )

sum += y;

}

cout << endl << "Sum = " << sum << endl;

}

void max()

{

double \*A;

int n = 0;

cout << "Input n, please:"; cin >> n;

A = new double[n];

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

{

cout << "Input A[" << i << "]:";

cin >> A[i];

}

int max = A[0];

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

{

if(A[i] > max) max = A[i];

}

cout << endl << "Max:" << max << endl;

delete A;

}

int factorial(int n)

{

if(n >= 0)

{

if(n == 0) return 1;

return n\*factorial(n-1);

}

else cout << "Error!!!!";

}

int Dobutok(int a, int b)

{

int y = 0, sum = 0;

for (int x = -20; x <= 20; x++)

{

sum \*= x;

}

cout << endl << "Sum = " << sum << endl;

return sum;

}

int MyFunction(int var, ...)

{

int rez = var;

va\_list ptr;

\_\_crt\_va\_start(ptr, var);

for (; var != '"'; var = \_\_crt\_va\_arg(ptr, char))

{

if (var == '+')

rez += \_\_crt\_va\_arg(ptr, int);

else if (var == '-')

rez -= \_\_crt\_va\_arg(ptr, int);

else if (var == '\*')

rez \*= \_\_crt\_va\_arg(ptr, int);

else if (var == '/')

rez /= \_\_crt\_va\_arg(ptr, int);

else if (var == '>>')

rez &= \_\_crt\_va\_arg(ptr, int);

else if (var == '%')

rez %= \_\_crt\_va\_arg(ptr, int);

}

\_\_crt\_va\_end(ptr);

return rez;

}

int Sum(int a, int b)

{

int y = 0, sum = 0;

for (int x = -20; x <= 20; x++)

{

sum += x;

}

cout << endl << "Sum = " << sum << endl;

return sum;

}

void myDefine()

{

int c ROVNO (5.5 MINUS 6.1 PLUS 1.8) DILENNJA 5 + 5 OSTACHA 2;

cout << "C:" << c << endl;

}

#if DEFINED

int main(int argc, char\*\* argv)

{

system("color f0");

int b = 4;

cout << endl << "Rezult: " << MyFunction(2, '+', 4, '"') << endl;

//max();

/\* cout << "8 ----"<< (8 ZSUV\_RIGHT 1) << endl;

cout << "12 ----"<< (12 ZSUV\_RIGHT 2) << endl;

int a = 0;

cin >> a;

cout << "Factorial:" << factorial(a) << endl;\*/

system("pause");

return NULL;

}

#else

int main(int argc, char\*\* argv)

{

system("color f0");

int a = 10;

int b = 4;

myDefine();

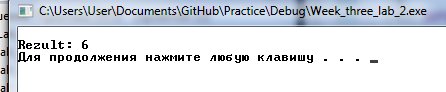
system("pause");

return NULL;

}

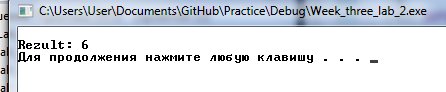
#endif

1. Тестування програми.



Програма працює вірно.

1. Результат виконання програми.



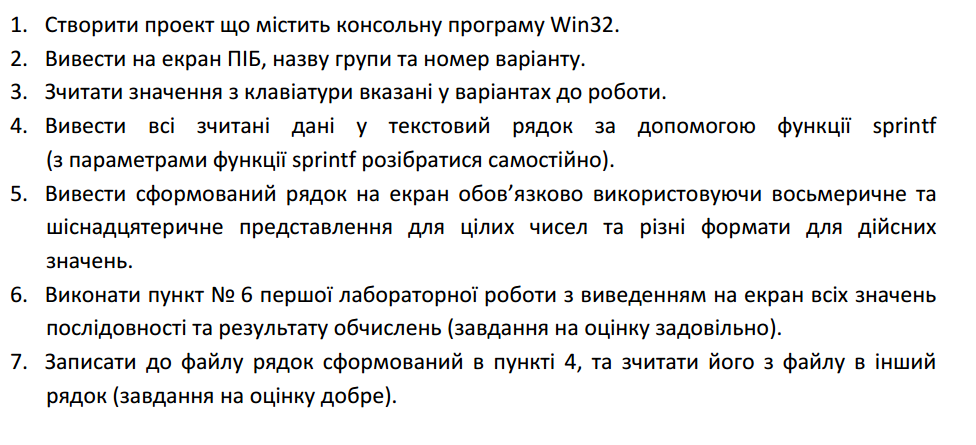
**Висновок:** я отримав практичні навики роботи з функціями користувача, навчився передавати аргументи в функцію, створювати та викликати власні функції, використовувати засоби умовної компіляції.

**Лабораторна робота №8**

**Тема:** базові типи даних та ввід-вивід.

**Мета роботи:** ознайомитись з функціями стандартної бібліотеки для введення даних з клавіатури та виведення на екран.

**Завдання:**





Хід роботи

1. Напишемо код моовою С++.

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string>

#include <stdio.h>

using namespace std;

int main()

{

char ch;

int integer;

short intSmall;

float fl;

string str;

ofstream fout;

ifstream fin;

/\*Вивід прирізвища та варіанту на екран \*/

cout << "Sapozhnyk Dmytro Olegovich, P-31, V-8." << endl << endl;

fout.open("Text.txt", ios\_base::out | ios\_base::trunc);

/\*Зчитуємо значення у змінні відповідно до таблиці \*/

cout << "Char: ";

cin >> ch;

fout << ch << endl;

cout << "Int: ";

cin >> integer;

fout << integer << endl;

cout << "Short: ";

cin >> intSmall;

fout << intSmall << endl;

cout << "Float: ";

cin >> fl;

fout << fl << endl;

cout << "String: ";

cin >> str;

fout << str << endl;

fout << endl << "----------------------------------" << endl << endl;

fout.close();

fout.open("Text.txt", ios\_base::out | ios\_base::app);

/\* Âèâåñòè âñ³ ç÷èòàí³ äàí³ ó òåêñòîâèé ðÿäîê çà äîïîìîãîþ ôóíêö³¿ sprintf \*/

cout << endl << "\t Output" << endl;

char buf[50];

sprintf\_s(buf, "Char: %c", ch);

printf\_s("%s\n", buf);

fout << buf << endl;

sprintf\_s(buf, "Int: %d", integer);

printf\_s("%s\n", buf);

fout << buf << endl;

sprintf\_s(buf, "Short: %hd", intSmall);

printf\_s("%s\n", buf);

fout << buf << endl;

sprintf\_s(buf, "Float: %.3f", fl);

printf\_s("%s\n", buf);

fout << buf << endl;

cout << "String: " << str << endl;

fout << "String: " << str << endl;

fout.close();

/\* Числа в різни форматах \*/

cout << endl << "\t Different number system" << endl;

sprintf\_s(buf, "Char(16): %x", ch);

printf\_s("%s\n", buf);

sprintf\_s(buf, "Int(8): %o", integer);

printf\_s("%s\n", buf);

sprintf\_s(buf, "Int(16): %x", integer);

printf\_s("%s\n", buf);

sprintf\_s(buf, "Short(8): %ho", intSmall);

printf\_s("%s\n", buf);

sprintf\_s(buf, "Short(16): %hx", intSmall);

printf\_s("%s\n", buf);

sprintf\_s(buf, "Float(exp): %.3e", fl);

printf\_s("%s\n", buf);

/\*ЗАвдання з попередньої лаби \*/

cout << endl << "\t Item 6" << endl;

int y = 0, sum = 0;

for (int x(-20); x <= 20; x++)

{

y = pow(x, 5) - 5\*pow(x, 3) + x + 37;

cout << "\t" << y << endl;

if ((y % 13) == 5)

sum += y;

}

cout << "Sum = " << sum << endl;

/\* зчитуємо зз файлу в змінні \*/

string strFile;

fin.open("Text.txt", ios\_base::in);

int i = 0;

while (fin.good())

{

fin.getline(buf, 50);

if (i++ > 9)

{

strFile += buf;

strFile += '\n';

}

}

fin.close();

cout << endl << "\t Read file" << endl << strFile << endl;

fin.open("Text.txt", ios\_base::in);

char \_ch;

fin.get(\_ch);

int \_integer;

fin >> \_integer;

short \_intSmall;

fin >> \_intSmall;

float \_fl;

fin >> \_fl;

string \_str;

fin >> \_str;

fin.close();

cout << " New var"

<< "Char: " << \_ch << endl

<< "Int: " << \_integer << endl

<< "Short: " << \_intSmall << endl

<< "Float: " << \_fl << endl

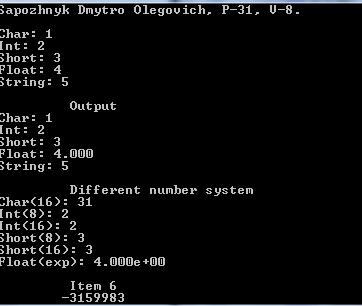
<< "String: " << \_str << endl;

system("pause");

return 0;

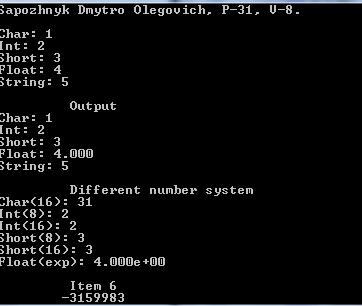
}

1. Тестування програми.



Програма працює вірно.

1. Результат виконання програми.



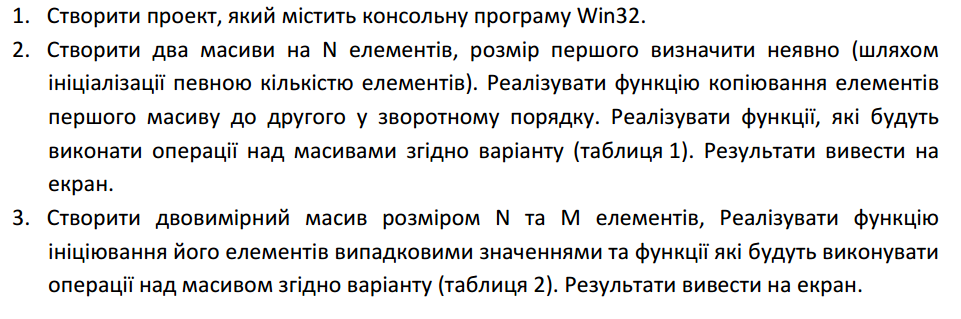
Висновок: я ознайомився з функціями стандартної бібліотеки для введення даних з клавіатури та виведення на екран.

**Лабораторна робота №9**

**Тема:** робота з одновимірними та багатовимірними масивами.

**Мета роботи:** метою лабораторної роботи є ознайомлення з використанням масивів, отримання практичних навиків роботи з одновимірними та багатовимірними масивами мовою С++.

**Завдання:**



**Хід роботи**

1. Напишемо код мовою С++.

#include <iostream>

#include <random>

#include <iomanip>

using namespace std;

void ArrOutput(const int \*arr, const int n)

{

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

printf("\t%5d", arr[i]);

}

void ArrBackCopy(const int\* arr1, int\* arr2, const int n)

{

for (int i = n - 1, j = 0; i >= 0 && j < n; i--, j++)

arr2[j] = arr1[i];

}

int ArrSumMin(const int \*arr, const int n)

{

int sum = 0;

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

if (arr[i] < 0)

sum += arr[i];

return sum;

}

int ArrSumDubut(const int \*A, const int n)

{

int sum = 0, j = 0;

int \*B = new int[n];

for (int i = 0; i < n; i += 2)

{

if (i == n - 1) B[j] = A[i];

else B[j] = A[i] \* A[i+1];

j++;

}

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

sum += B[i];

delete B;

return sum;

}

int\*\*& ArrCreate(const int n, const int m)

{

int \*\*arr = new int\*[n];

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

arr[i] = new int[m];

return arr;

}

void ArrDelete(int \*\*&arr, const int n)

{

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

delete[] arr[i];

delete[] arr;

}

void ArrRecordRand(int \*\*arr, const int n, const int m)

{

random\_device random;

uniform\_int\_distribution<int> rd(-10, 10);

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

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

arr[i][j] = rd(random);

}

void ArrOutput(int \*\*arr, const int n, const int m)

{

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

{

cout << '\t';

ArrOutput(arr[i], m);

cout << endl;

}

}

void ArrZeroOverDiagonal(int \*\*A, const int n, const int m)

{

int i, S = 0, flag = 0;

cout << endl << "Nuber string not null:";

for (i = 0; i < n; i++)

{

flag = 0;

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

if (A[i][j] != 0) flag++;

if (flag == m) cout << "\t" << i + 1 << "\t";

}

}

void ArrSumCol(int \*\*arr, const int n, const int m)

{

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

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

if (((i + j) % 4) == 0) arr[i][j] = 0;

}

int main()

{

system("color f0");

const int arr1[] = { -1, 2, 3, -6, 7, -2};

const int N = 6;

int arr2[N];

cout << "Array 2: ";

ArrBackCopy(arr1, arr2, N);

ArrOutput(arr2, N);

cout << endl << endl;

cout << "Suma min elem: " << ArrSumMin(arr2, N) << endl;

cout << "Suma dobutkiv elem: " << ArrSumDubut(arr2, N) << endl;

int n, m;

cout << "Enter size array [n, m]: ";

cin >> n >> m;

int \*\*arr3 = ArrCreate(n, m);

ArrRecordRand(arr3, n, m);

cout << endl << "Array Rand:" << endl;

ArrOutput(arr3, n, m);

cout << endl << "Array zero over diagonal:" << endl;

ArrZeroOverDiagonal(arr3, n, m);

ArrSumCol(arr3, n, m);

cout << endl << "Array null:" << endl;

ArrOutput(arr3, n, m);

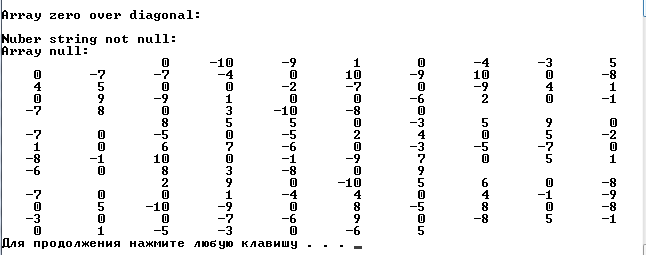
ArrDelete(arr3, n);

system("pause");

return 0;

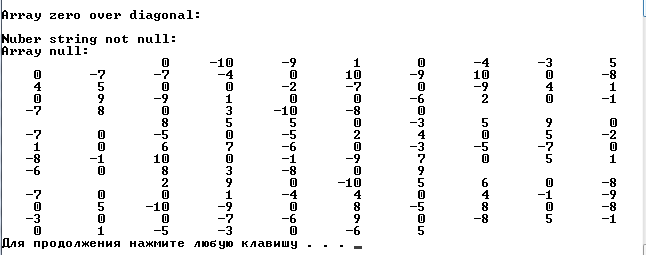
}

1. Тестування програми.



Програма працює вірно.

1. Результат виконання програми.



**Висновок:** я ознайомився з використанням масивів, отримання практичних навиків роботи з одновимірними та багатовимірними масивами мовою С++.

**Лабораторна робота №10**

**Тема:** Рядки. Функції по роботі з рядками.

**Мета роботи:** метою даної лабораторної роботи є отримання навиків по роботі з рядками шляхом використання засобів стандартної бібліотеки для обробки рядків мовою С++.

**Завдання:**

1. Створити проект, який містить консольну програму Win32.

2. Зчитати з клавіатури два довільні рядки, обрати та виконати наступні операції над цими рядками з використанням функцій стандартної бібліотеки:

a. на оцінку задовільно – копіювання рядків, копіювання n символів одного рядка в інший, склеювання рядків, додання n символів одного рядка до іншого, порівняння рядків, порівняння n символів одного рядка з іншим;

b. на оцінку добре – пошук першого входження визначеного аргументом символу до рядку, пошук першого входження визначеного аргументом рядку до іншого рядку.

3. Без використання функцій стандартної бібліотеки виконати завдання згідно варіантудо роботи, таблиця 1 (завдання на оцінку відмінно).

Хід роботи

1. Напишемо код мовою С++.

#include <iostream>

#include <cstring>

using namespace std;

void find(char \* str, int pos)

{

int flag = 0;

for (int i = 0; i < strlen(str); i++)

{

if (str[i] == ' ' || str[i] == '!' || str[i] == '?' || str[i] == ',' || str[i] == '.' || str[i] == ':' || str[i] == ';') continue;

for (int j = i; j < strlen(str); j++)

{

if (str[j] == ' ' || str[j] == '!' || str[j] == '?' || str[j] == ',' || str[j] == '.' || str[j] == ':' || str[j] == ';') continue;

if (str[j + 1] == ' ' || str[j + 1] == '\0' || str[j + 1] == '!' || str[j + 1] == '?' || str[j + 1] == ',' || str[j + 1] == '.' || str[j + 1] == ':' || str[j + 1] == ';')

{

flag++;

if (flag == pos) cout << "Index pos: " << i << endl;

i = j + 1;

break;

}

}

}

}

#define ON 0

int main()

{

const int MAX = 50;

char str1[MAX], str2[MAX];

#if ON

cout << "Enter two strings:" << endl;

cout << "str1 > ";

cin.get(str1, MAX);

cin.ignore(MAX, '\n');

cin.clear();

cout << "str2 > ";

cin.get(str2, MAX);

cin.ignore();

cin.clear();

cout << endl << "--------------------------------------------" << endl;

char str3[MAX];

strcpy\_s(str3, str1);

cout << endl << "str3 = str1: " << str3 << endl;

int num;

cout << endl << "Enter number of letter for copy (str2 in str3): ";

cin >> num;

strncpy\_s(str3, str2, num);

cout << "str3 = str2(n): " << str3 << endl;

strcat\_s(str1, str2);

cout << endl << "str1 = str1 + str2: " << str1 << endl;

cout << endl << "Enter number of letter for addition (str1 in str3): ";

cin >> num;

strncat\_s(str3, str1, num);

cout << "str3 = str3 + str1(n): " << str3 << endl;

cout << endl << "Comparison str1, str2: ";

if (strcmp(str1, str2) < 0)

cout << "str1 < str2" << endl;

else if (strcmp(str1, str2) > 0)

cout << "str1 > str2" << endl;

else cout << "str1 == str2" << endl;

int temp;

cout << endl << "Enter number of letter for check (str3, str1(n)): ";

cin >> num;

cout << endl << "Comparison str3(n), str1(n): ";

if (temp = strncmp(str3, str1, num) < 0)

cout << "str3(n) < str1(n)" << endl;

else if (temp = strncmp(str3, str1, num) > 0)

cout << "str3(n) > str1(n)" << endl;

else cout << "str3(n) == str1(n)" << endl;

char let;

cout << endl << "Enter letter for search (str2): ";

cin >> let;

cin.ignore();

cin.clear();

cout << "Index letter: ";

if (strchr(str2, let) != NULL)

cout << strchr(str2, let) - str2 << endl;

else cerr << "Letter don\'t found" << endl;

char str4[MAX];

cout << endl << "Enter string for search (str1): ";

cin.get(str4, MAX);

cin.ignore();

cin.clear();

cout << "Index string: ";

if (strstr(str1, str4) != NULL)

cout << strstr(str1, str4) - str1 << endl;

else cerr << "String don\'t found" << endl;

cout << endl << "------------------------------------------------------" << endl;

#else

cout << "Enter two string:" << endl;

cout << "str1 > ";

cin.get(str1, MAX);

cin.ignore(MAX, '\n');

cin.clear();

int pos;

cout << "Enter starn and number words:";

cin >> pos;

find(str1, pos);

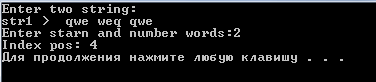
#endif

system("pause");

return 0;

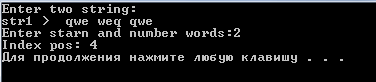
}

1. Тестування прогами.



Програм працює вірно.

1. Результат виконання програми.



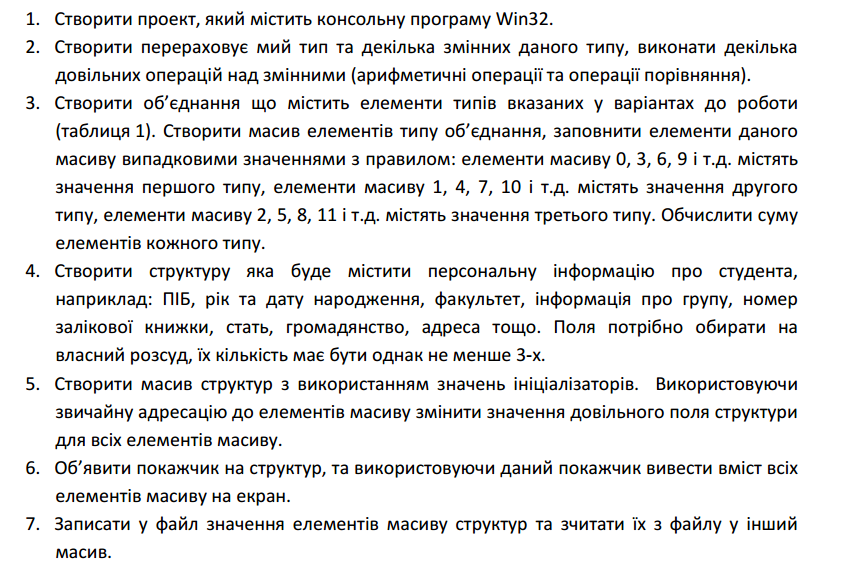
**Висновок:** я отримав навики по роботі з рядками шляхом використання засобів стандартної бібліотеки для обробки рядків мовою С++.

**Лабораторна робота №11**

**Тема:** Перераховуваний тип. Об'єднання.Структури.

**Мета роботи:** Отримати практичні навики роботи з Перераховуваними типами, об'єднаннями та структурами з використанням мови С++.

**Завдання:**





**Хід роботи**

1. Напишемо код мовою С++.

#include <iostream>

#include <fstream>

#include <string>

#include <random>

using namespace std;

enum days\_of\_week { Mon, Tue, Wed, Thu, Fri, Sat, Sun };

union MyUnion

{

char Ch;

int Int;

short Short;

float Float;

string\* String;

};

struct Date

{

int day;

int mount;

int year;

};

struct Student

{

string name;

Date dateBirth;

string profession;

string group;

int numberGradebook;

};

void StudentInput(Student& st)

{

ofstream fout("Text.txt", ios\_base::app);

cout << "Enter name: ";

getline(cin, st.name);

fout << st.name << endl;

cout << "Enter date of birth (DD MM YYYY): ";

cin >> st.dateBirth.day >> st.dateBirth.mount >> st.dateBirth.year;

fout << st.dateBirth.day << ' ' << st.dateBirth.mount << ' ' << st.dateBirth.year << endl;

cin.ignore();

cin.clear();

cout << "Enter profession: ";

getline(cin, st.profession);

fout << st.profession << endl;

cout << "Enter group: ";

cin >> st.group;

fout << st.group << endl;

cout << "Enter number of Gradebook: ";

cin >> st.numberGradebook;

fout << st.numberGradebook << endl << endl;

cin.ignore();

cin.clear();

fout.close();

}

void StudentOutput(const Student\* st, const int n)

{

for (int i = 0; i < n; i++, st++)

{

cout << "Name: " << st->name << endl;

cout << "Date of birth: " << st->dateBirth.day << '/' << st->dateBirth.mount << '/' << st->dateBirth.year << endl;

cout << "Profession: " << st->profession << endl;

cout << "Group: " << st->group << endl;

cout << "Number of Gradebook: " << st->numberGradebook << endl;

cout << endl << "--------------------------------------------------------------" << endl << endl;

}

}

Student\*& StRead(const int N)

{

Student \*st = new Student[N];

ifstream fin("Text.txt");

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

{

getline(fin, st[i].name);

fin >> st[i].dateBirth.day >> st[i].dateBirth.mount >> st[i].dateBirth.year;

fin.ignore();

fin.clear();

getline(fin, st[i].profession);

getline(fin, st[i].group);

fin >> st[i].numberGradebook;

fin.ignore();

fin.ignore();

fin.clear();

}

fin.close();

return st;

}

int main()

{

system("color f0");

days\_of\_week day1, day2;

day1 = Tue;

day2 = Thu;

int diff = day1 - day2;

cout << "day1: Mon\tday2: Thu" << endl;

cout << "The difference in days: " << diff << endl;

if (day1 < day2)

cout << "\tday1 < day2" << endl;

else cout << "\tday1 > day2" << endl;

cout << endl << "-------------------------------------------" << endl << endl;

const int MAX = 15;

MyUnion myTypes[MAX];

random\_device random;

long long llInt = 0;

unsigned long long ullInt = 0;

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

{

if (i == 0 || i % 3 == 0)

{

uniform\_int\_distribution<int> rd('a', 'x');

myTypes[i].Ch = rd(random);

cout << "Char: " << myTypes[i].Ch << endl;

}

else if (i == 1 || ((i - 1) % 3) == 0)

{

uniform\_int\_distribution<int> rd(-2147483647, 2147483647);

myTypes[i].Int = rd(random);

cout << "Int: " << myTypes[i].Int << endl;

llInt += myTypes[i].Int;

}

else if (i == 2 || ((i - 2) % 3) == 0)

{

uniform\_int\_distribution<short> rd(0, 255);

myTypes[i].Short = rd(random);

cout << "Short: " << myTypes[i].Short << endl;

ullInt += myTypes[i].Short;

}

if (i != 0 && (i + 1) % 3 == 0)

cout << endl;

}

cout << "Sum Int: " << llInt << endl

<< "Sum Short: " << ullInt << endl << endl;

/\*myTypes->Short = random();

cout << "Short: " << myTypes->Short << endl;

\*/

myTypes->Float = random();

cout << "Float: " << myTypes->Float << endl;

myTypes->String = new string;

\*myTypes->String = "My name is Dmytro.";

cout << "String: " << \*myTypes->String << endl;

delete myTypes->String;

system("pause");

system("cls");

ofstream clear("Text.txt", ios\_base::trunc);

clear.close();

const int STMAX = 2;

Student st[STMAX];

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

{

StudentInput(st[i]);

cout << endl << "--------------------------------------------------" << endl;

}

system("cls");

Student \*pst = st;

StudentOutput(pst, STMAX);

cout << "Read file:" << endl;

Student \*stFile = StRead(STMAX);

StudentOutput(stFile, STMAX);

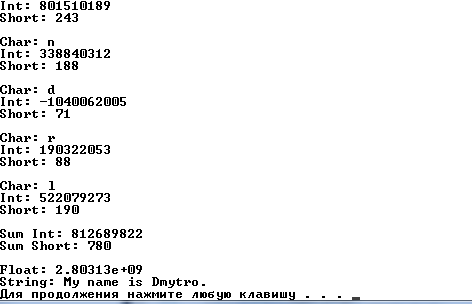
delete[] stFile;

system("pause");

return 0;

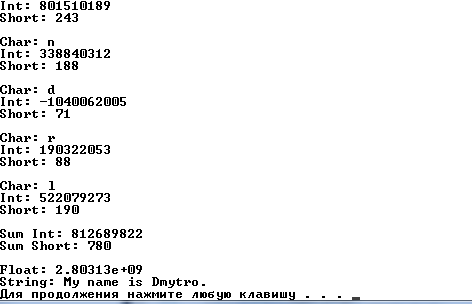
}

1. Тестування програми.



Програма працює вірно.

1. Результат виконання програми.



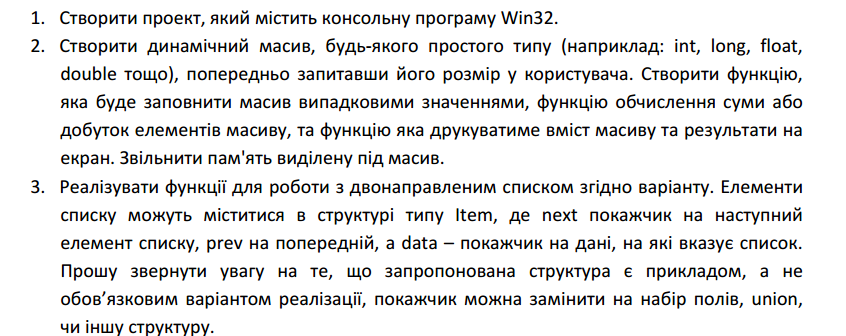
**Висновок:** я отримав практичні навики роботи з перераховуваними типами, об'єднаннями та структурами з використанням мови С++.

**Лабораторна робота №12**

**Тема:** Динамічний розподіл пам'яті. Покажчики та їх використання.

**Мета роботи:** Метою лабораторної роботи є ознайомлення з методами динамічного розподілу пам’яті, отримання практичних навиків роботи з функціями виділення та звільнення пам’яті мовою С++.

**Завдання:**





**Хід роботи**

1. Напишемо код мовою С++.

#include <iostream>

#include <random>

using namespace std;

struct Item

{

int data;

Item \*back, \*next;

};

void InputData(int &var)

{

cout << "Enter number: ";

cin >> var;

}

void OutputDate(const Item \*ptr)

{

cout << ptr->data << endl;

}

inline bool EmptyListInfo(const Item \*list)

{

if (list != NULL)

return false;

cout << "\t List empty" << endl;

return true;

}

Item\* CreateItem()

{

Item \*ptr = new Item;

InputData(ptr->data);

ptr->back = NULL;

ptr->next = NULL;

return ptr;

}

Item\* AddToTheStart(Item \*first)

{

Item \*p = CreateItem();

if (first)

first->back = p;

p->next = first;

return p;

}

Item\* DeleteElem(Item \*ptr)

{

Item \*elem = ptr->next;

delete ptr;

return elem;

}

/\* Очищення списку \*/

void DeleteList(Item \*&list)

{

if (EmptyListInfo(list))

return;

else cout << "\t List cleaned" << endl;

while (list != NULL)

list = DeleteElem(list);

}

/\* Додавання елементу в кінець списку \*/

void AddToTheEnd(Item \*ptr)

{

while (ptr->next != NULL)

ptr = ptr->next;

ptr->next = CreateItem();

ptr->next->back = ptr;

}

/\* Отримання покажчика на i–й елемент списку \*/

Item\* PtrElementN(Item \*list, const int n)

{

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

{

if (list == NULL)

return NULL;

list = list->next;

}

return list;

}

/\* Отримання номеру елементу списку за покажчиком на елемент \*/

int NElementPtr(const Item \*list, Item \*elem)

{

int i = 0;

while (list != elem)

{

list = list->next;

i++;

}

return i + 1;

}

/\* Вставка елемента в список перед визначеним елементом \*/

void AddElem(Item \*elem)

{

Item \*el = CreateItem();

el->back = elem->back;

el->next = elem;

elem->back->next = el;

elem->back = el;

}

int main(int argc, char \*\*argv)

{

system("color f0");

int n;

cout << "Enter size array: ";

cin >> n;

int \*arr = new int[n];

ArrRand(arr, n);

cout << "Array: ";

ArrOutput(arr, n);

cout << endl << "Sum array: " << ArrSum(arr, n) << endl;

delete[] arr;

system("pause");

system("cls");

Item \*list = NULL;

SwichMenu(list);

return 0;

}

1. Тестування програми.



Програми працює вірно.

1. Результат виконання програми.



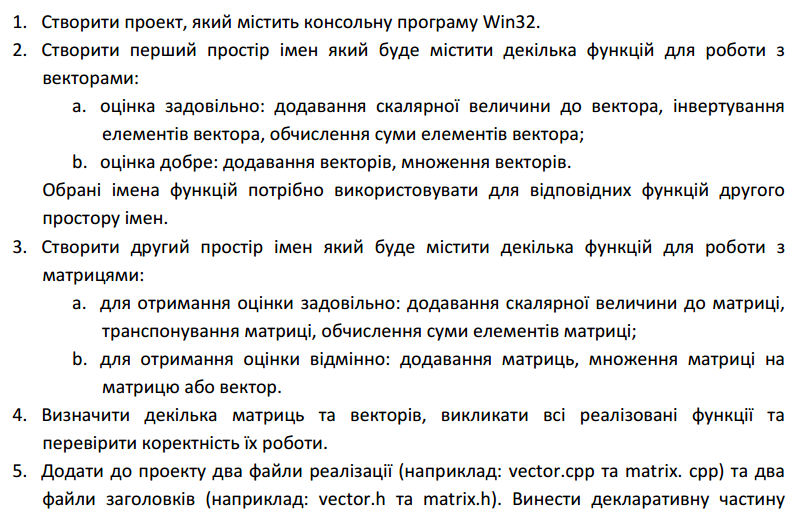
**Висновок:** я ознайомився з методами динамічного розподілу пам’яті, отримання практичних навиків роботи з функціями виділення та звільнення пам’яті мовою С++.

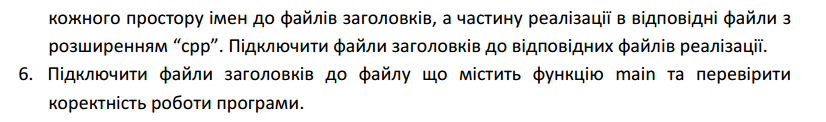
**Лабораторна робота №13**

**Тема:** Простір імен. Компонування.

**Мета роботи:** Метою лабораторної роботи є ознайомлення з механізмом відображення логічного групування (так званим простором імен) та створенням програм які складаються з набору декількох вхідних файлів.

**Завдання:**





**Хід роботи**

1. Напишемо код мовою С++.

#include <iostream>

#include <string>

using namespace std;

void erase( char \* str)

{

for(int i = 0; i < strlen(str); i++)

{

if(str[i] == ' ' || str[i] == '!' || str[i] == '?' || str[i] == ',' || str[i] == '.' || str[i] == ':' || str[i] == ';') continue;

for(int j = i; j < strlen(str); j++)

{

if(str[j] == ' ' || str[j] == '!' || str[j] == '?' || str[j] == ',' || str[j] == '.' || str[j] == ':' || str[j] == ';') continue;

if(str[j + 1] == ' ' || str[j + 1] == '\0' || str[j + 1] == '!' || str[j + 1] == '?' || str[j + 1] == ',' || str[j + 1] == '.' || str[j + 1] == ':' || str[j + 1] == ';')

{

if(str[i] == str[j])

{

for(int l = i; l < strlen(str); l++)

{

str[l] = str[l + (j - i) + 1];

}

//cout << endl << "TEST string:" << str << endl;

i = j = 0;

break;

}

else

{

i = j + 1;

break;

}

}

}

}

}

bool is\_it(char\* str)

{

return str[0] == 'q' || str[0] =='Q' || str[0] == 'w' || str[0] == 'W' || str[0] == 'r' || str[0] == 'R' || str[0] == 't' || str[0] == 'T' || str[0] == 'p' || str[0] == 'P' || str[0] == 's' || str[0] == 'S' || str[0] == 'd' || str[0] == 'D' || str[0] == 'f' || str[0] == 'F' || str[0] == 'g' || str[0] == 'G' || str[0] == 'h' || str[0] == 'H' || str[0] == 'l' || str[0] == 'L' || str[0] == 'k' || str[0] == 'K' || str[0] == 'z' || str[0] == 'Z' || str[0] == 'x' || str[0] == 'X' || str[0] == 'c' || str[0] == 'C' || str[0] == 'v' || str[0] == 'V' || str[0] == 'b' || str[0] == 'B' || str[0] == 'n' || str[0] == 'N' || str[0] == 'm' || str[0] == 'M';

}

int NumOfDigits(char\* str)

{

int count = 0;

cout << endl << "Len string:" << strlen(str)<< endl;

for(int i = 0; i < strlen(str); i++)

{

if(isdigit(str[i]))

count++;

}

return count;

}

int main()

{

char numbers[999];

char numbers1[999];

cout << "Input string: ";

cin.get(numbers, 1000, '\n');

for( int i = 0; i < 999; i++)

numbers1[i] = numbers[i];

cout << endl << "New string:" << numbers1 << endl;

erase(numbers1);

cout << endl << "Erases string:" << numbers1 << endl;

cout << "Numbers in string: " << NumOfDigits(numbers) << endl;

cout << "Word: ";

for(char \*p = strtok(numbers, " "); p; p = strtok(0, " "))

{

if(is\_it(p))

{

cout << p << " ";

}

}

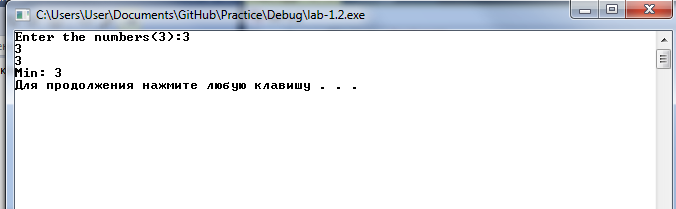
//cout << endl << "New string:" << str << endl;\*/

system("pause");

return 0;

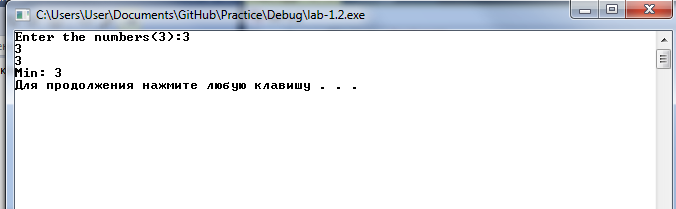
}

1. Тестування програми.



Програма працює вірно.

1. Результат виконання прогами.



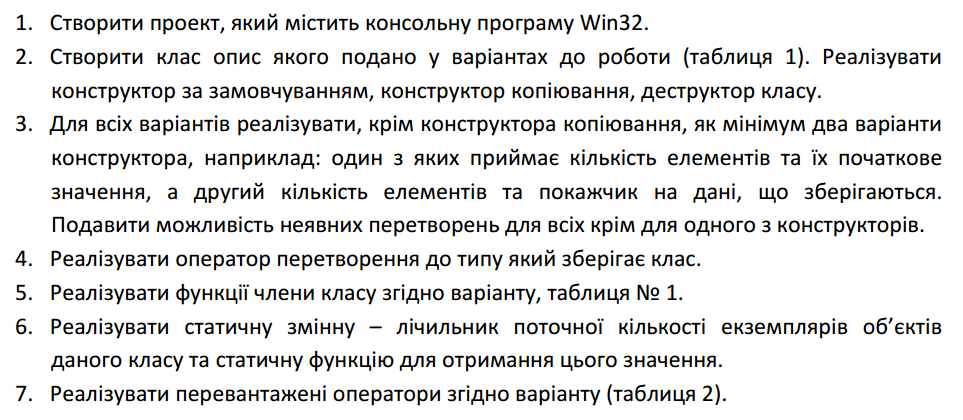
**Висновок:** я ознайомився з механізмом відображення логічного групування (так званим простором імен) та створенням програм які складаються з набору декількох вхідних файлів.

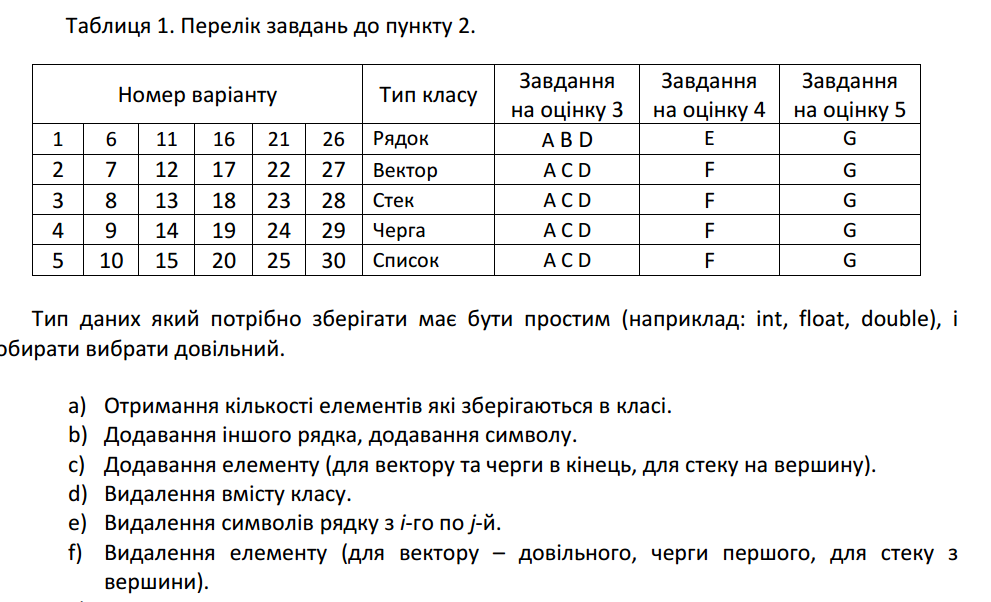
**Лабораторна робота №14**

**Тема:** Класи. Спеціальні функції-члени класу. Перевантаження операцій. Дружні функції.

**Мета роботи:** Метою лабораторної роботи є ознайомлення з реалізацією основ об’єктно-орієнтованого програмування мовою C++.

**Завдання:**





**Хід роботи**

1. Напишемо код мовою С++.

#include <string>

#include <iostream>

using namespace std;

template<class T>

class Stack

{

public:

Stack() :tail(0), head(0), size(0)

{

}

~Stack()

{

while (head)

{

tail = head->next;

delete head;

head = tail;

}

}

void push(T val)

{

Node\* Temp;

Temp = new Node;

Temp->elem = val;

if (tail == 0)

{

tail = Temp;

}

else

{

Temp->next = tail;

tail = Temp;

}

size++;

}

T top()

{

if (tail == 0)

{

cout << endl <<"Stack is empty!"<< endl;

return 0;

}

return tail->elem;

}

void clear()

{

while (tail != NULL)

{

Node\* delptr = tail;

tail = tail->next;

delete delptr;

}

cout << endl <<"Stack is empty!"<< endl;

}

T pop()

{

if (tail == 0)

{ cout << endl <<"Stack is empty!"<< endl;

return 0; }

Node\* delptr = tail;

cout << endl << tail->elem << endl;

tail = tail->next;

delete delptr;

//return tail->elem;

}

void print()

{

if (tail == 0)

{

cout << endl <<"Stack is empty!"<< endl;

}

for (Node\* ptr = tail; ptr != 0; ptr = ptr->next)

{

std::cout << ptr->elem << ' ';

}

std::cout << '\n';

}

void getIndex(int q)

{

if (q > size)

{

cout << endl << "Error!" << endl;

return;

}

if (tail == 0)

{

cout << endl <<"Stack is empty!"<< endl;

return;

}

int flag = 0;

for (Node\* ptr = tail; ptr != 0; ptr = ptr->next)

{

flag++;

if(flag == q) cout << "Elem index " << q << " : " << ptr->elem << endl;

}

}

void SwichMenu();

inline void OutputMenu()

{

cout << "\t Select an action" << endl

<< "1 - Clean the stack;" << endl

<< "2 - Adding an element to the end of the stack;" << endl

<< "4 - Getting a pointer to the i-th element of the stack;" << endl

<< "5 - Get stack item ;" << endl

<< "6 - Print the list;" << endl

<< "7 - Pop the list;" << endl

<< "0 - Exit." << endl;

}

private:

struct Node

{

Node() :elem(0), next(0)

{

}

Node\* next;

T elem;

};

Node\* head;

Node\* tail;

int size;

};

template <typename T>

void Stack<T>::SwichMenu()

{

int action;

while (true)

{

OutputMenu();

cout << endl << "Select an action: ";

cin >> action;

switch (action)

{

;

case 1:

clear();

break;

case 2:

T a;

cout << "INtput element: ";

cin >> a;

push(a);

break;

case 4:

cout << "Enter number element: ";

int elem;

cin >> elem;

if (elem <= 0)

{

cerr << "\t Incorrect input" << endl;

break;

}

getIndex(elem);

break;

case 5:

cout << "Element: " << top() << endl;

break;

case 6:

{

print();

break;

}

case 7:

{

pop();

break;

}

case 0: return;

default:

cerr << "\t Incorrect input" << endl;

break;

}

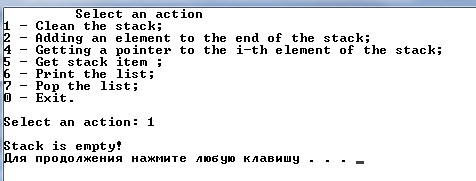
system("pause");

system("cls");

}

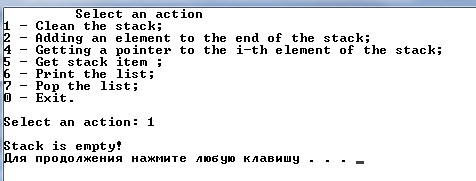
}

1. Тестування програми.



Програма працює вірно.

1. Результат виконання програми.



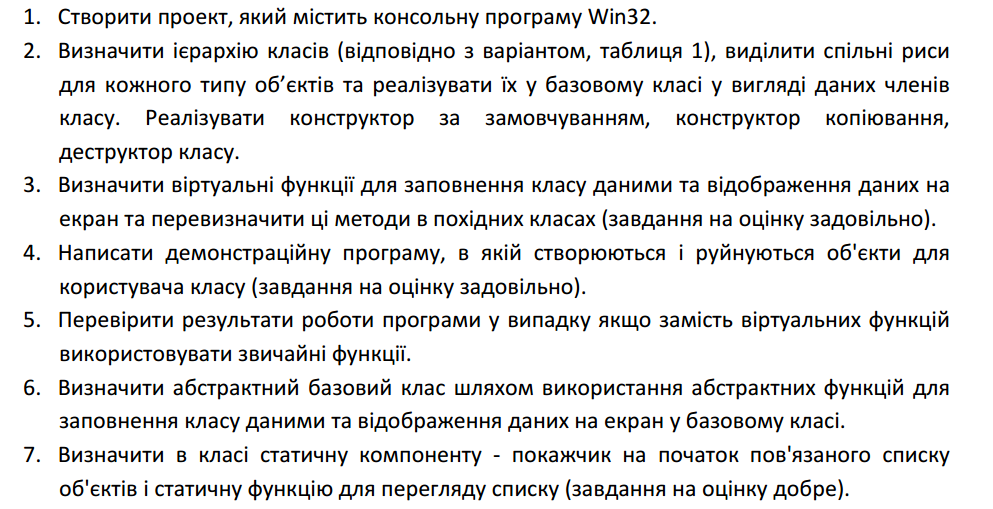
**Висновок:** я ознайомився з реалізацією основ об’єктно-орієнтованого програмування мовою C++.

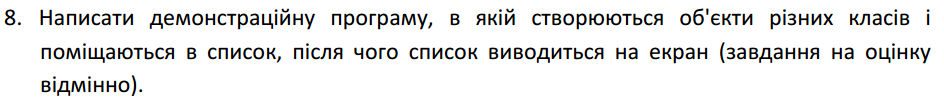
**Лабораторна робота №15**

**Тема:** Класи. Успадкування. Віртуальні функції та абстрактні класи.

**Мета роботи:** Метою лабораторної роботи є ознайомлення з механізмами успадкування, віртуальних функцій та абстрактних класів в мові C++.

**Завдання:**







**Хід роботи**

1. Напишемо код мовою С++.

**Лабораторна робота №16**