**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет   
імені Петра Могили**

**Факультет комп’ютерних наук**

**Кафедра інженерії програмного забезпечення**

**ЗВІТ**

*з індивідуального завдання № 1*

Дисципліна "Основи програмування"

Спеціальність: **інженерія програмного забезпечення**

121-ІЗ.01-108.1710809

***Cтудент*** *\_\_\_\_\_\_\_\_\_\_\_\_\_****І.М.Зурілов***

*(підпис)*

*\_\_12.12.17 \_\_*

*(дата)*

***Викладач****\_\_\_\_\_\_\_\_\_\_\_\_****Є.О.Давиденко***

*(підпис)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

**Миколаїв 2017**

**Вимоги до інтерфейсу**:

Кожне завдання має бути реалізовано в **окремій функції**. Функції мають розташовуватись у файлах:

**ind\_.h – прототипи функцій;**

**indf.срр – реалізація\_функцій;**

**ind.срр – реалізація основної програми.**

Основна програма має містити тільки команди для завантаження вхідних даних, виклик описаних функцій та вивід отриманих результатів.

Всі функції повинні отримувати вхідні значення (з основної програми) в якості своїх параметрів-аргументів та повертати результат (до основної програми).

**Завдання №1.**

Скласти програму обчислення значень змінних за заданими розрахунковими формулами. На екран вивести значення вхідних даних і результати обчислень, супроводжуючи вивід найменуваннями виведених змінних.

Кожну формулу реалізувати в окремій функції.

9.Ввести з клавіатури цілі w, y та дійсні x, z, v. Розрахувати

;

;

.

**Завдання №2 Умовний оператор**

Обчислити значення функції **Y**

|  |  |
| --- | --- |
| 9. |  |

**Завдання №3. Масиви (вектор) –3 бали**

Описати статичний масив.

Реалізувати 3 способи ініціалізації масиву (користувач має сам обрати спосіб, зробити меню):

* сформувати масив (***n*=20** елементів) за вказаним правилом;

Визначити:

|  |  |  |
| --- | --- | --- |
| **Парні варіанти** | Суму елементів масиву | Максимальне значення масиву, його індекс |
| **Непарні варіанти** | Середньоарифметичне елементів масиву | Мінімальне значення масиву, його індекс |

* ввести значення з клавіатури (виконати завдання за варіантом);
* заповнити масив псевдовипадковими числами (виконати завдання за варіантом).

Результат записати до текстового файлу.

|  |  |  |
| --- | --- | --- |
| 9. | Масив дійсних чисел: | Визначити два сусідніх елементи масиву, сума яких найбільш близька до числа *R* |

**Завдання №4. Файли – 3 бали**

Результат записати до іншого текстового файлу.

9. Задано 2 текстових файли. Записати у третій файл тільки ті рядки, які є і у першому і у другому файлах.

cout << "Ener firsts integer w, y, alpha and beta; then double x, z, v:" << endl;

cin >> w >> y >> alpha >> beta >> x >> z >> v;

double exercise\_1A(int w\_1, double x\_1, double z\_1, double v\_1)

Початок

if (3\*sqrt (x\_1) >= 0 &&( abs (x\_1 - pow(z\_1, 3.0/2)) - 15\*x\_1\*x\_1 ) != 0 &&2\*v\_1 != 0 &

pow (1 / 2 \* v\_1, 1.0/4) >= 0 && sqrt (x\_1) >= 0 &&

1 - sqrt (x\_1) !=0 && sqrt(2\*w\_1) != 0)

**true false**

A\_1 = ( 1 + 3 \* sqrt(x\_1) ) / (abs(x\_1 - pow(z\_1, 3.0 / 2)) - 15 \* x\_1 \* x\_1)+ 1 / (( ( pow(1 / 2 \* v\_1, 1.0 / 4) + sqrt (4 \* w\_1 \* w\_1) / 1 - sqrt(x\_1) )- sqrt(2 \* w\_1) ) );

cout << "You've got a false variable";

double exercise\_1B(int w\_1, int y\_1, double x\_1, double z\_1, double v\_1)

if (x\_1\*w\_1 + z\_1 != 0 && w\_1 != 0 && 3\*v\_1 + y\_1 != 0)

**true false**

cout << "Some variable(s) is(are) wrong";

B\_1 = ( 1 / x\_1\*w\_1 + z\_1 ) + ( ( pow(x\_1, 5) \* y\_1 ) / w\_1 ) + abs(17 \* x\_1 \* y\_1) + ( ( z\_1 / 3 \* v\_1 + y\_1 ) \* pow(10, 5) \* y\_1);

double exercise\_1C(int alpha\_1, int beta\_1)

C\_1 = (pow(cos(3 \* alpha\_1), 2) \* cos(beta\_1))- 3 \* pow((sin((30 \* PI / 180) - 3 \* alpha\_1)), 3)+ sin(7 \* beta\_1);

cout << "C is " << exercise\_1C(alpha, beta) << endl;

cout << "B is " << exercise\_1B(w, y, x, z, v) << endl;

cout << "A is " << exercise\_1A(w, x, z, v) << endl;

Вихід

**Рис. 1 – Блок-схема розв’язку задачі 1**

double exercise\_2(double a\_1, double b\_1, double x\_1)

cout << "Enter a, b and x :";

cin >> a >> b >> x;

Початок

**true false**

cout << "There was used condition 1" << endl;

cout << "There was used condition 2" << endl;

if (x\_1 <= a\_1)

**default**

y = 11.3 + 3.3 \* cos (3.6 \* x\_1 - PI/4);

y = pow( ( cos(x\_1) + sin (x\_1) ) , cos(2 \* x\_1) );

cout << "There was used condition 3" << endl;

y = 2.8 \* log10(3 \* x\_1) + 8 \* pow(x\_1, 3.1) + 4.3 \* pow(x\_1, 2.7) + 1.1;

Вихід

cout << "Y is " << exercise\_2(a, b, x) << endl;

**Рис. 2 – Блок-схема розв’язку задачі 2**

cout << "\nInput 1 - default , 2 - input elements by keyboard, 3 - random numbers : ";cin >> variant;

Початок

cout << "Firsts, enter R :";

cin >> R;

switch (variant)

case 3: exercise\_3\_2(R); break;

case 1: exercise\_3(R); break;

default: exercise\_3(R); break;

case 2: exercise\_3\_1(R); break;

Вихід

**Рис. 3 – Блок-схема розв’язку задачі 3**

// default та case 1 обчислення, де відомий лише R; массив заповнюється автоматично

cout << "\nArray by the rule : A[i] = i^3 + 11\*i - (i - 3)(i + 4) is ";

void exercise\_3(double R\_1)

case 1: exercise\_3(R); break;

// Знаходимо мінімальне значення масиву, його індекс

**true**

ofstream example("ex3.txt");

for (int i = 0; i < arr.size(); i++)

**false false**

for (int i = 0; i < arr.size(); i++)

**true**

arr[i] = (i\*i\*i) + 11\*i - (i - 3)\*(i + 4);

cout << arr[i] << " ";

**true false**

if(arr[i] < min)

min = arr[i]; indCounter++;

cout << "\nThe least element of the array is " << min << ". \nIt's index is " << indCounter << endl;

example << "\nThe least element of the array is " << min << ". \nIt's index is " << indCounter << endl;

// Знаходимо середньоарифметичне значення елементів масиву

cout << "\nAverage of array elements is " << average << endl;

average = sumOfEl / arr.size();

for (int i = 0; i < arr.size(); i++)

sumOfEl += arr[i];

Вихід

example << "\nAverage of array elements is " << average << endl;

**Рис. 3.1 – Блок-схема case 1 та default**

// Визначити два сусідніх елементи масиву, сума яких найбільш близька до числа R (ввід елементів з клавіатури)

for (int i = 0; i < arr1.size(); i++)

cout << "\nEnter your variables : ";

Ofstream example("ex3.txt");

void exercise\_3\_1(double R\_1)

**true**

cin >> arr1[i];

cout << " ";

for (int i = 0; i < arr1.size(); i++)

**true false**

diff = R\_1 - (arr1[i] + arr1[i + 1]);

if (diff > 0 && diff <= 5)

**false**

example.close();

**true**

Вихід

cout << "\nSum of " << i << " and " << i+1 << " is close to R" << endl;

example << "\nSum of " << i << " and " << i+1 << " is close to R" << endl;

**Рис. 3.2 – Блок-схема case 2**

// Визначити два сусідніх елементи масиву, сума яких найбільш близька до числа R (псевдовипадкові числа)

srand(time(NULL));

ofstream example("ex3.txt");

void exercise\_3\_2(double R\_1)

**true**

for (int i = 0; i < arr2.size(); i++)

cout << "\nEnter your variables : ";

arr2[i] = rand() % 20 - 0;;

cout << arr2[i] << " ";

for (int i = 0; i < arr2.size(); i++)

**false**

example.close();

**true**

Вихід

if (diff > 0 && diff <= 5)

diff = R\_1 - (arr2[i] + arr2[i + 1]);

cout << "\nSum of " << i << " and " << i+1 << " is close to R" << endl;

example << "\nSum of " << i << " and " << i+1 << " is close to R" << endl;

**Рис. 3.3 – Блок-схема case 3**

ifstream file1("file1.txt"); ifstream file2("file2.txt"); ofstream file3("file3.txt");

void exercise\_4()

Початок

cout << s1 << endl;

file2 >> s2;

cout << s2 << endl;

cout << s1 << endl;

file1 >> s1;

file3 << s1;

strcat\_s(s1, s2);

Вихід

file1.close(); file2.close(); file3.close();

**Рис. 4 – Блок-схема розв’язку задачі 4**

**Програмний код**

**ind\_.h**

// Файл прототипів ф-цій

//Завдання 1 (головне)

double exercise\_1A(int, double, double, double); // Розрахунок А

double exercise\_1B(int,int, double, double, double); // Розрахунок В

double exercise\_1C(int, int); // Розрахунок C

// ----------------------------------------------------

//Завдання 2 (головне)

double exercise\_2(double, double, double); // Розрахунок Y

// ----------------------------------------------------

//Завдання 3 (головне)

void exercise\_3(double);

void exercise\_3\_1(double);

void exercise\_3\_2(double);

// ----------------------------------------------------

//Завдання 4

void exercise\_4();

**ind.cpp**

// Індивідуальна робота Зурілов І. гр. 108 Варіант 9

#include "stdafx.h"

#include <iostream>

#include <cmath>

#include <vector>

#include <fstream>

#include <cstring>

#include "ind\_.h" // Файл прототипів ф-цій

using namespace std;

int main()

{

/\*

Завдання 1 (головне)

Скласти програму обчислення значень змінних за заданими розрахунковими формулами.

На екран вивести значення вхідних даних і результати обчислень,

супроводжуючи вивід найменуваннями виведених змінних.

Кожну формулу реалізувати в окремій функції.

--------------------

Завдання 1.9

Ввести з клавіатури цілі w, y та дійсні x, z, v.

Розрахувати А, В, С

\*/

int w, y, alpha, beta; // alpha, beta - градусна міра кутів для прикл. 3

double x, z, v;

cout << "Ener firsts integer w, y, alpha and beta; then double x, z, v:" << endl;

cin >> w >> y >> alpha >> beta >> x >> z >> v;

cout << "A is " << exercise\_1A(w, x, z, v) << endl;

cout << "B is " << exercise\_1B(w, y, x, z, v) << endl;

cout << "C is " << exercise\_1C(alpha, beta) << endl;

// ----------------------------------------------------

/\*

Завдання 2 (головне)

Обчислити значення функції Y за допомогою умовного оператора

\*/

double a, b, x;

cout << "Enter a, b and x :";

cin >> a >> b >> x;

cout << "Y is " << exercise\_2(a, b, x) << endl;

// ----------------------------------------------------

/\*

Завдання 3 (головне)

Описати статичний масив.

Реалізувати 3 способи ініціалізації масиву (користувач має сам обрати спосіб,

зробити меню):

— сформувати масив (n=20 елементів) за вказаним правилом;

Визначити: (непарні варіанти)

середньоарифметичне елементів масиву, мінімальне значення

масиву, його індекс

— ввести значення з клавіатури (виконати завдання за варіантом);

— заповнити масив псевдовипадковими числами (виконати завдання за варіантом).

Результат записати до текстового файлу.

--------------------

9. Масив дійсних чисел: A[i] = i^3 + 11\*i - (i - 3)(i + 4) ; n = 20 (розмірність масиву)

Визначити два сусідніх елементи масиву, сума яких

найбільш близька до числа R.

\*/

int variant;

double R;

cout << "Firsts, enter R :";

cin >> R;

cout << "\nInput 1 - default , 2 - input elements by keyboard, 3 - random numbers : ";

cin >> variant;

switch (variant)

{

case 1: exercise\_3(R); break;

case 2: exercise\_3\_1(R); break;

case 3: exercise\_3\_2(R); break;

default: exercise\_3(R); break;

}

// ----------------------------------------------------

/\*

Завдання 4

9. Задано 2 текстових файли. Записати у третій файл тільки

ті рядки, які є і у першому і у другому файлах.

\*/

void exercise\_4();

system("pause");

return 0;

}

**indf.cpp**

// Файл реалізації ф-цій

#include "stdafx.h"

#include <iostream>

#include <cmath>

#include <vector>

#include <fstream>

#include <time.h>

#include <cstring>

#include "ind\_.h" // Файл прототипів ф-цій

using namespace std;

const double PI = 3.14;

// Завдання 1

double exercise\_1A(int w\_1, double x\_1, double z\_1, double v\_1)

{

double A\_1;

if (

3\*sqrt (x\_1) >= 0 && // Квадратний корінь не може бути від'ємним

( abs (x\_1 - pow(z\_1, 3.0/2)) - 15\*x\_1\*x\_1 ) != 0 && // Знаменник не дорівнює 0

2\*v\_1 != 0 && // Знаменник не дорівнює 0

pow (1 / 2 \* v\_1, 1.0/4) >= 0 && // Парний корінь не може бути від'ємним

sqrt (x\_1) >= 0 && // Квадратний корінь не може бути від'ємним

1 - sqrt (x\_1) !=0 && // Знаменник не дорівнює 0

sqrt(2\*w\_1) != 0 // Квадратний корінь не може бути від'ємним

)

{

A\_1 = ( 1 + 3 \* sqrt(x\_1) ) / (abs(x\_1 - pow(z\_1, 3.0 / 2)) - 15 \* x\_1 \* x\_1)

+ 1 / (( ( pow(1 / 2 \* v\_1, 1.0 / 4) + sqrt (4 \* w\_1 \* w\_1) / 1 - sqrt(x\_1) )

- sqrt(2 \* w\_1) ) );

}

else

{

cout << "You've got a false variable";

}

return A\_1;

}

double exercise\_1B(int w\_1, int y\_1, double x\_1, double z\_1, double v\_1)

{

double B\_1;

if (

x\_1\*w\_1 + z\_1 != 0 && // Знаменник не дорівнює 0

w\_1 != 0 && // Знаменник не дорівнює 0

3\*v\_1 + y\_1 != 0 // Знаменник не дорівнює 0

)

{

B\_1 = ( 1 / x\_1\*w\_1 + z\_1 ) + ( ( pow(x\_1, 5) \* y\_1 ) / w\_1 )

+ abs(17 \* x\_1 \* y\_1) + ( ( z\_1 / 3 \* v\_1 + y\_1 ) \* pow(10, 5) \* y\_1 );

}

else

{

cout << "Some variable(s) is(are) wrong";

}

return B\_1;

}

double exercise\_1C(int alpha\_1, int beta\_1)

{

double C\_1;

C\_1 = (pow(cos(3 \* alpha\_1), 2) \* cos(beta\_1))

- 3 \* pow((sin((30 \* PI / 180) - 3 \* alpha\_1)), 3)

+ sin(7 \* beta\_1);

return C\_1;

}

// ----------------------------------------------------

// Завдання 2

double exercise\_2(double a\_1, double b\_1, double x\_1)

{

double y;

if (x\_1 <= a\_1)

{

cout << "There was used condition 1" << endl;

y = pow( ( cos(x\_1) + sin (x\_1) ) , cos(2 \* x\_1) );

}

else if (x\_1 > a\_1 && x\_1 < b\_1)

{

cout << "There was used condition 2" << endl;

y = 11.3 + 3.3 \* cos (3.6 \* x\_1 - PI/4);

}

else

{

cout << "There was used condition 3" << endl;

y = 2.8 \* log10(3 \* x\_1) + 8 \* pow(x\_1, 3.1) + 4.3 \* pow(x\_1, 2.7) + 1.1;

}

return y;

}

// ----------------------------------------------------

//Завдання 3

int indCounter = 0; // Індекс мін. знач. массиву

double average, sumOfEl = 0, diff; // diff - різниця між R та суммою 2 сусід. ел. массиву

void exercise\_3(double R\_1) // default та case 1 обчислення, де відомий лише R; массив заповнюється автоматично

{

vector <double> arr(20);

cout << "\nArray by the rule : A[i] = i^3 + 11\*i - (i - 3)(i + 4) is ";

for (int i = 0; i < arr.size(); i++)

{

arr[i] = (i\*i\*i) + 11\*i - (i - 3)\*(i + 4);

cout << arr[i] << " ";

}

ofstream example("ex3.txt");

// Знаходимо мінімальне значення масиву, його індекс

double min = arr[0];

for (int i = 0; i < arr.size(); i++)

{

if(arr[i] < min)

{

min = arr[i];

indCounter++;

}

}

cout << "\nThe least element of the array is " << min << ". \nIt's index is " << indCounter << endl;

example << "\nThe least element of the array is " << min << ". \nIt's index is " << indCounter << endl;

// Знаходимо середньоарифметичне значення елементів масиву

for (int i = 0; i < arr.size(); i++)

{

sumOfEl += arr[i];

}

average = sumOfEl / arr.size();

cout << "\nAverage of array elements is " << average << endl;

example << "\nAverage of array elements is " << average << endl;

example.close();

}

// ---------------------------------------------------------------

void exercise\_3\_1(double R\_1)

{

vector <double> arr1(5);

ofstream example("ex3.txt");

cout << "\nEnter your variables : ";

for (int i = 0; i < arr1.size(); i++)

{

cin >> arr1[i];

cout << " ";

}

// Визначити два сусідніх елементи масиву, сума яких найбільш близька до числа R

for (int i = 0; i < arr1.size(); i++)

{

diff = R\_1 - (arr1[i] + arr1[i + 1]);

if (diff > 0 && diff <= 5)

{

cout << "\nSum of " << i << " and " << i+1 << " is close to R" << endl;

example << "\nSum of " << i << " and " << i+1 << " is close to R" << endl;

}

}

example.close();

}

// ---------------------------------------------------------------

void exercise\_3\_2(double R\_1)

{

vector <double> arr2(5);

ofstream example("ex3.txt");

srand(time(NULL));

cout << "\nEnter your variables : ";

for (int i = 0; i < arr2.size(); i++)

{

arr2[i] = rand() % 20 - 0;;

cout << arr2[i] << " ";

}

// Визначити два сусідніх елементи масиву, сума яких найбільш близька до числа R

for (int i = 0; i < arr2.size(); i++)

{

diff = R\_1 - (arr2[i] + arr2[i + 1]);

if (diff > 0 && diff <= 5)

{

cout << "\nSum of " << i << " and " << i + 1 << " is close to R" << endl;

example << "\nSum of " << i << " and " << i + 1 << " is close to R" << endl;

}

}

example.close();

}

// ----------------------------------------------------

//Завдання 4

void exercise\_4()

{

char s1[100], s2[100];

ifstream file1("file1.txt"); ifstream file2("file2.txt"); ofstream file3("file3.txt");

file1 >> s1;

cout << s1 << endl;

file2 >> s2;

cout << s2 << endl;

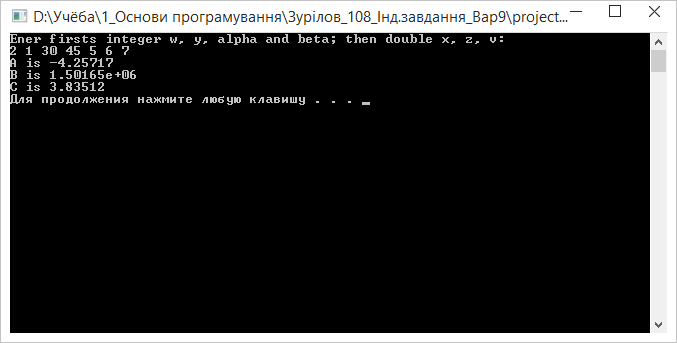
strcat\_s(s1, s2);

cout << s1;

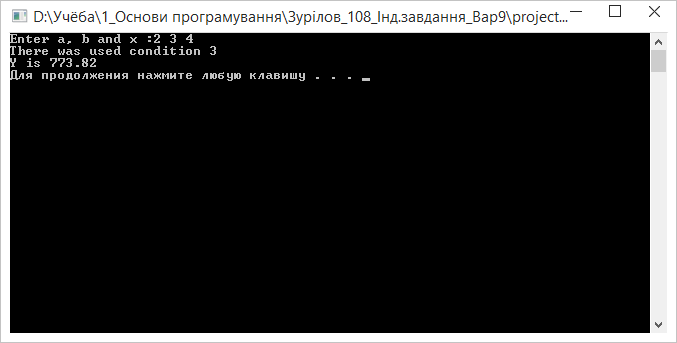
file3 << s1;

file1.close(); file2.close(); file3.close();

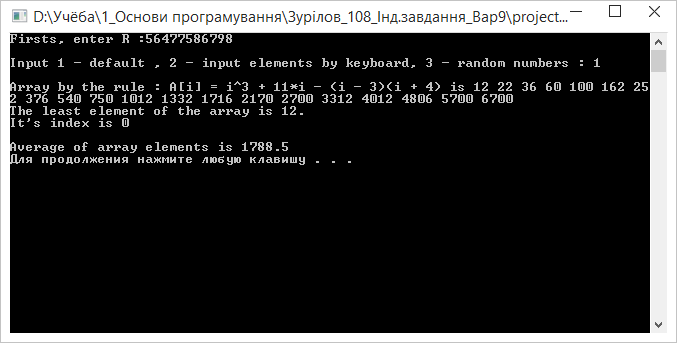
}



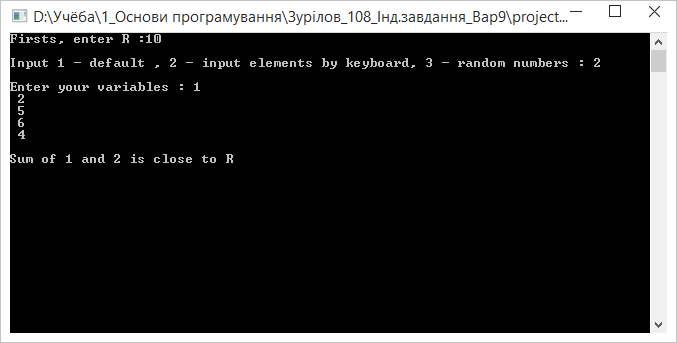
**Виконання програми до завд. 1**



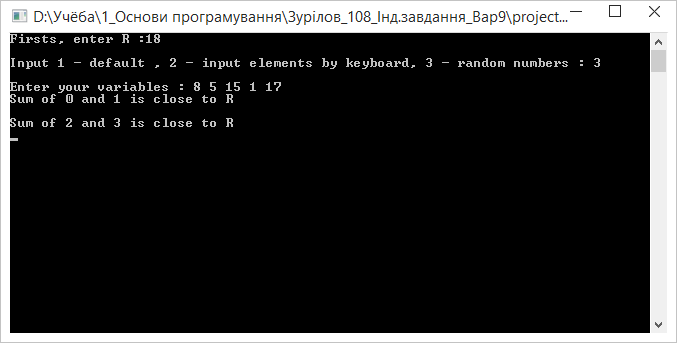
**Виконання програми до завд. 2**



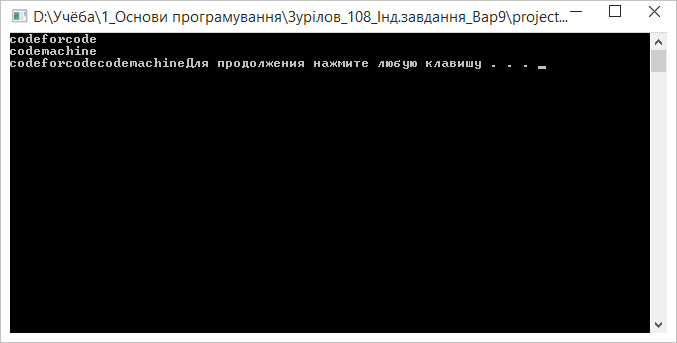
**Виконання програми до завд. 3.1**



**Виконання програми до завд. 3.2**



**Виконання програми до завд. 3.3**



**Виконання програми до завд. 4**

**Висновок :** під час індивідуальної роботи були використані знання, здобуті протягом навчального семестру та навички роботи з математичними функціями, циклами з умовою, операторами умови, масивами чисел та символів,символами, рядками, функціями та файлами. Були розроблені схеми розв’язку типових задач за допомогою мови С++, зокрема передача значень іншому масиву. Блок-схема розв’язку, алгоритм та виконання занесені у документ.