МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КОЛЕДЖ ЧЕРНІВЕЦЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА

**ЗВІТ**

про навчальну практику

студент(а/ки) II курсу, спеціальності

«Комп`ютерні науки»

204(214) групи

*Дрищука Степана Олеговича*

Період практики з “20.07” 2020р. по “07.08” 2020 р.

Керівник практики від коледжу:

викл. Ковдриш В.В. \_\_\_\_\_\_\_\_\_\_

/підпис/

Оцінка:\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

/підпис/

ЧЕРНІВЦІ, 2020

ТАБЛИЦЯ ОЦІНЮВАННЯ

|  |  |  |
| --- | --- | --- |
| Завдання | Оцінка за завдання | Підпис керівника практики |
| Завдання 1. |  |  |
| Завдання 2. |  |  |
| Завдання 3. |  |  |
| Завдання 4. |  |  |
| Завдання 5. |  |  |
| Завдання 6. |  |  |
| Завдання 7. |  |  |
| Завдання 8. |  |  |
| Завдання 9. |  |  |
| Завдання 10. |  |  |
| Завдання 11. |  |  |
| Завдання 12. |  |  |
| Оформлення звіту | |  |
| Захист звіту | |  |
| Підсумкова оцінка за навчальну практику | |  |

Зміст

[Завдання 1 7](#_Toc47344403)

[ПОСТАНОВКА ЗАДАЧІ 8](#_Toc47344404)

[Опис завдання 8](#_Toc47344405)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 8](#_Toc47344406)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 8](#_Toc47344407)

[Висновок 8](#_Toc47344408)

[Додаток(а) 8](#_Toc47344409)

[Додаток(б) 8](#_Toc47344410)

[Завдання 2 9](#_Toc47344411)

[ПОСТАНОВКА ЗАДАЧІ 10](#_Toc47344412)

[Опис завдання 10](#_Toc47344413)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 10](#_Toc47344414)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 10](#_Toc47344415)

[Висновок 10](#_Toc47344416)

[Додаток(а) 10](#_Toc47344417)

[Додаток(б) 10](#_Toc47344418)

[Завдання 3 11](#_Toc47344419)

[ПОСТАНОВКА ЗАДАЧІ 12](#_Toc47344420)

[Опис завдання 12](#_Toc47344421)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 12](#_Toc47344422)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 12](#_Toc47344423)

[Висновок 12](#_Toc47344424)

[Додаток(а) 12](#_Toc47344425)

[Додаток(б) 12](#_Toc47344426)

[Завдання 4 13](#_Toc47344427)

[ПОСТАНОВКА ЗАДАЧІ 14](#_Toc47344428)

[Опис завдання 14](#_Toc47344429)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 14](#_Toc47344430)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 14](#_Toc47344431)

[Висновок 14](#_Toc47344432)

[Додаток(а) 14](#_Toc47344433)

[Додаток(б) 14](#_Toc47344434)

[Завдання 5 15](#_Toc47344435)

[ПОСТАНОВКА ЗАДАЧІ 16](#_Toc47344436)

[Опис завдання 16](#_Toc47344437)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 16](#_Toc47344438)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 16](#_Toc47344439)

[Висновок 16](#_Toc47344440)

[Додаток(а) 16](#_Toc47344441)

[Додаток(б) 16](#_Toc47344442)

[Завдання 6 17](#_Toc47344443)

[ПОСТАНОВКА ЗАДАЧІ 18](#_Toc47344444)

[Опис завдання 18](#_Toc47344445)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 18](#_Toc47344446)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 18](#_Toc47344447)

[Висновок 18](#_Toc47344448)

[Додаток(а) 18](#_Toc47344449)

[Додаток(б) 18](#_Toc47344450)

[Завдання 7 19](#_Toc47344451)

[ПОСТАНОВКА ЗАДАЧІ 20](#_Toc47344452)

[Опис завдання 20](#_Toc47344453)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 20](#_Toc47344454)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 20](#_Toc47344455)

[Висновок 20](#_Toc47344456)

[Додаток(а) 20](#_Toc47344457)

[Додаток(б) 20](#_Toc47344458)

[Завдання 8 21](#_Toc47344459)

[ПОСТАНОВКА ЗАДАЧІ 22](#_Toc47344460)

[Опис завдання 22](#_Toc47344461)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 22](#_Toc47344462)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 22](#_Toc47344463)

[Висновок 22](#_Toc47344464)

[Додаток(а) 22](#_Toc47344465)

[Додаток(б) 22](#_Toc47344466)

[Завдання 9 23](#_Toc47344467)

[ПОСТАНОВКА ЗАДАЧІ 24](#_Toc47344468)

[Опис завдання 24](#_Toc47344469)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 24](#_Toc47344470)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 24](#_Toc47344471)

[Висновок 24](#_Toc47344472)

[Додаток(а) 24](#_Toc47344473)

[Додаток(б) 24](#_Toc47344474)

[Завдання 10 25](#_Toc47344475)

[ПОСТАНОВКА ЗАДАЧІ 26](#_Toc47344476)

[Опис завдання 26](#_Toc47344477)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 26](#_Toc47344478)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 26](#_Toc47344479)

[Висновок 26](#_Toc47344480)

[Додаток(а) 26](#_Toc47344481)

[Додаток(б) 26](#_Toc47344482)

[Завдання 11 27](#_Toc47344483)

[ПОСТАНОВКА ЗАДАЧІ 28](#_Toc47344484)

[Опис завдання 28](#_Toc47344485)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 28](#_Toc47344486)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 28](#_Toc47344487)

[Висновок 28](#_Toc47344488)

[Додаток(а) 28](#_Toc47344489)

[Додаток(б) 28](#_Toc47344490)

[Завдання 12 29](#_Toc47344491)

[ПОСТАНОВКА ЗАДАЧІ 30](#_Toc47344492)

[Опис завдання 30](#_Toc47344493)

[ТЕОРЕТИЧНІ ВІДОМОСТІ 30](#_Toc47344494)

[СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ 30](#_Toc47344495)

[Висновок 30](#_Toc47344496)

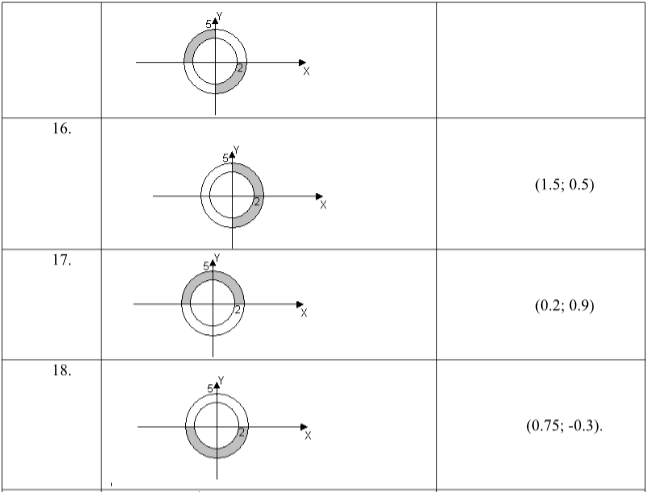
[Додаток(а) 30](#_Toc47344497)

[Додаток(б) 30](#_Toc47344498)

# Завдання 1

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

**Завдання:** Створити алгоритм та написати програму для розвязку наступної задачі: перевірку попадання точки ( яка буде вводитись вручну із клавіатури) в заштриховану область. №15

## ТЕОРЕТИЧНІ ВІДОМОСТІ

**Умовний перехід** — конструкція [мови програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), яка дозволяє виконувати різні дії залежно від [булевого](https://uk.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D1%96%D1%87%D0%BD%D0%B8%D0%B9_%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) значення умови вказаної програмістом.

Найбільш часто умовний перехід має дві стадії: на першій відбувається порівняння між собою деяких величин, що визначають умову переходу, а на другій виконується сам перехід.

Необхідність коректної обробки умовних переходів накладає серйозний відбиток на логіку роботи сучасних конвеєрних процесорів[[1]](https://uk.wikipedia.org/wiki/%D0%A3%D0%BC%D0%BE%D0%B2%D0%BD%D0%B8%D0%B9_%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D1%96%D0%B4#cite_note-1). Умовні переходи можуть виконуватися двома способами. Виконувані переходи часто змінюють значення лічильника команд [процесора](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80) на обчислене значення адреси переходу. Невиконувані — додають до значення лічильника команд число, рівне довжині поточної команди в байтах, для переходу до виконання наступної команди. Неправильне визначення типу умовного переходу може призводити до виникнення суттєвих затримок в роботі конвеєра і відповідно до великої втрати продуктивності [комп'ютера](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80).

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Розрахунок довжини радіуса та перевірка правильності умові

## Висновок

Я навчився використовувати умовний оператор if

## Додаток(а)

#include <iostream>

#include <math.h>

using namespace std;

float x, y;

int main()

{

std::cout << "Vvedit x:" << std::endl;

std::cin >> x;

std::cout << "Vvedit y:" << std::endl;

std::cin >> y;

if(sqrt((x\*x + y\*y)) > 2 && sqrt((x\*x + y\*y)) < 5 && (x \* y < 0)){

cout<<"To4ka prisutnya";

} else {

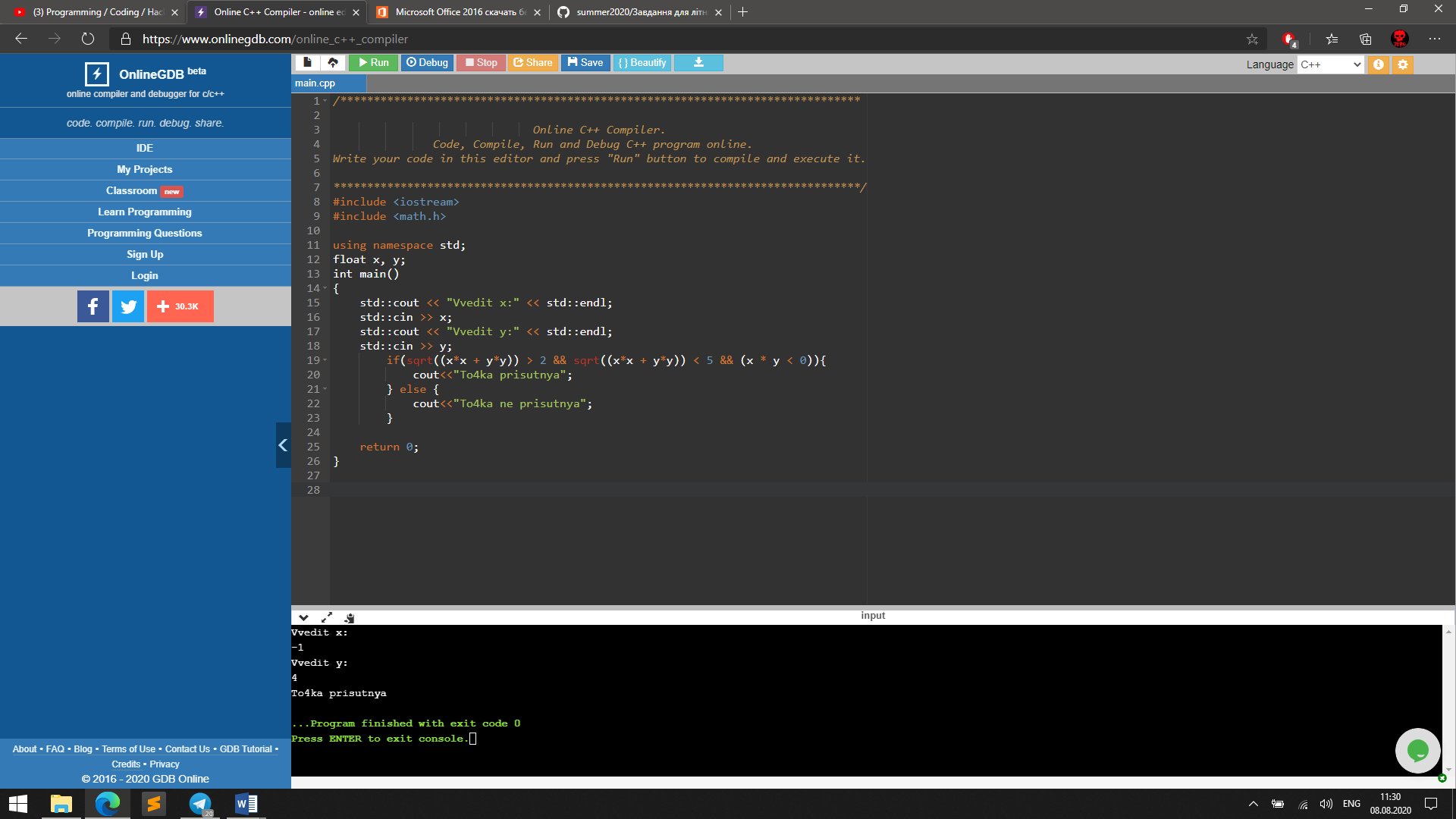
cout<<"To4ka ne prisutnya";

}

return 0;

}

## Додаток(б)



# Завдання 2

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

15. Всі від'мні елементи заданого масиву L(11) розділити на максимальний

елемент цього масиву. Вивести максимальний елемент, початковий та перетворений масиви.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

**Умовний перехід** — конструкція [мови програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), яка дозволяє виконувати різні дії залежно від [булевого](https://uk.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D1%96%D1%87%D0%BD%D0%B8%D0%B9_%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) значення умови вказаної програмістом.

Найбільш часто умовний перехід має дві стадії: на першій відбувається порівняння між собою деяких величин, що визначають умову переходу, а на другій виконується сам перехід.

Необхідність коректної обробки умовних переходів накладає серйозний відбиток на логіку роботи сучасних конвеєрних процесорів[[1]](https://uk.wikipedia.org/wiki/%D0%A3%D0%BC%D0%BE%D0%B2%D0%BD%D0%B8%D0%B9_%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D1%96%D0%B4#cite_note-1). Умовні переходи можуть виконуватися двома способами. Виконувані переходи часто змінюють значення лічильника команд [процесора](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80) на обчислене значення адреси переходу. Невиконувані — додають до значення лічильника команд число, рівне довжині поточної команди в байтах, для переходу до виконання наступної команди. Неправильне визначення типу умовного переходу може призводити до виникнення суттєвих затримок в роботі конвеєра і відповідно до великої втрати продуктивності [комп'ютера](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80).

**Цикл** ([лат.](https://uk.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B8%D0%BD%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *cyclus*, від [грец.](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B5%D1%86%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) kuklos — [круг](https://uk.wikipedia.org/wiki/%D0%9A%D1%80%D1%83%D0%B3)) — завершена [послідовність](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%81%D0%BB%D1%96%D0%B4%D0%BE%D0%B2%D0%BD%D1%96%D1%81%D1%82%D1%8C) функціонування матеріально-технічних компонентів об'єкта ([явища](https://uk.wikipedia.org/wiki/%D0%AF%D0%B2%D0%B8%D1%89%D0%B5)) в просторі та часі. Коло подій, явищ, частин.

Цикл — сукупність взаємозв'язаних явищ, процесів, що створюють закінчене коло розвитку протягом якогось проміжку часу (напр., Ц. виробничий, Ц. парової машини, двигуна внутрішнього згоряння). Замкнений Ц. — технологічний процес, за яким певний продукт переробки багаторазово повертається до вихідної операції і бере участь у наступному циклі переробки. У гірничій справі це, напр., процес подрібнення руди в замкненому Ц., багаторазове використання технологічної води в Ц. збагачення після її реґенерації (прояснення).

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Перебрати масив, знайти максимальний елемент та розділити всі елементи менші 0 на нього

## Висновок

Я навчився працювати з масивами, освоїв одновимірні цикли, та перерахунок

## Додаток(а)

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

float L[11]; int i,j, n=11;

printf("\nArray:");

printf("\n");

srand(time(NULL));

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

{

L[i] = rand()%20-5;

printf("%.2f ", L[i]); } printf("\n");

float max=L[0];

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

if (L[i]> max)

max =L[i];}

printf("\tMaximum = %.2f", max); printf("\n");

printf("\nNew array:");

printf("\n");

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

{if(L[i]<0){

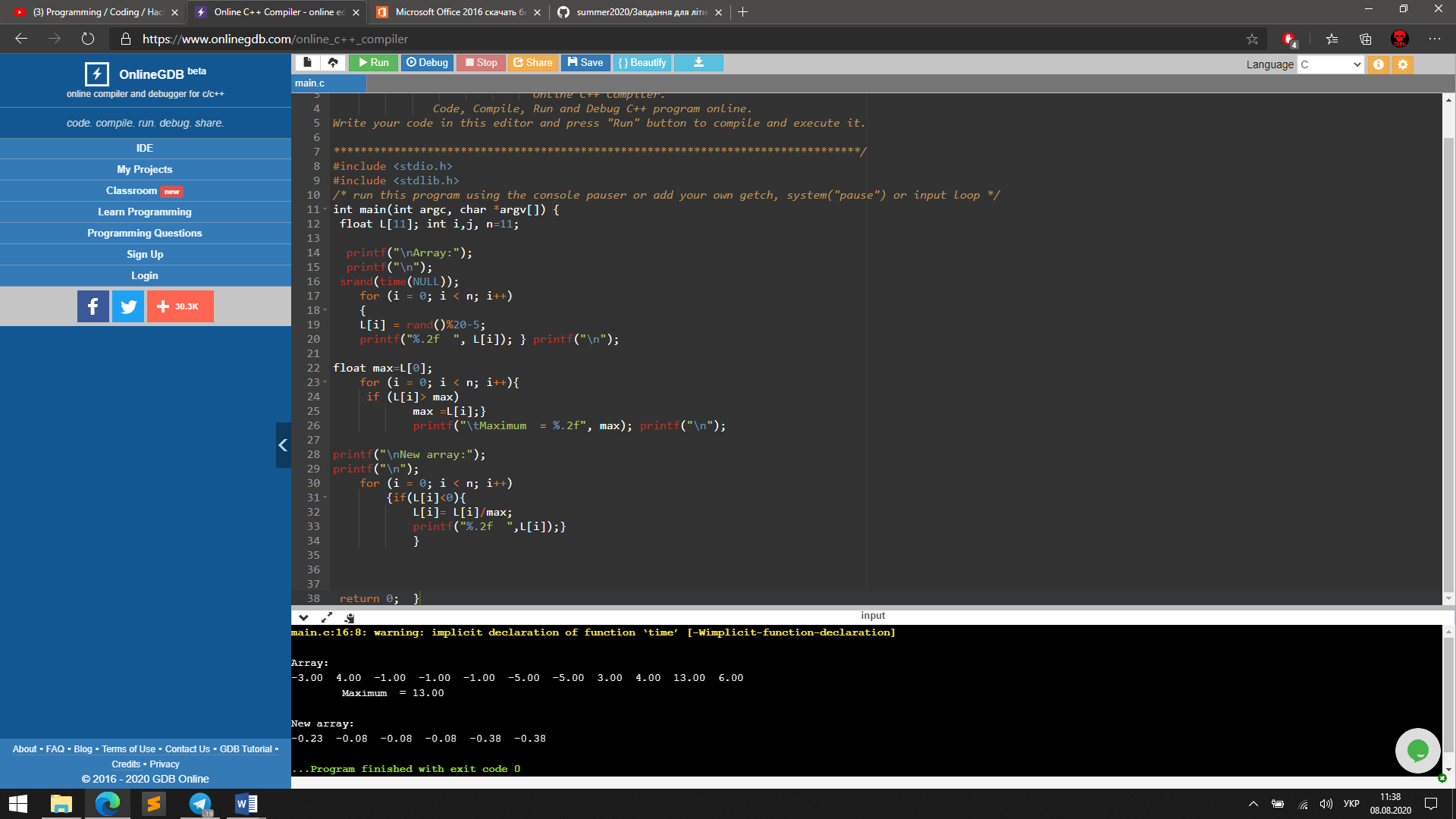
L[i]= L[i]/max;

printf("%.2f ",L[i]);}

}

return 0; }

## Додаток(б)



# Завдання 3

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

15. Задано два масиви A(8) та B(8), якi упорядкованi за спаданням. Визначити кiлькicть елементiв, що спiвпадають у двох масивах та їх значення. Вивести вихiднi масиви, елементи, що спiвпадають та їх кількість.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

**Умовний перехід** — конструкція [мови програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), яка дозволяє виконувати різні дії залежно від [булевого](https://uk.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D1%96%D1%87%D0%BD%D0%B8%D0%B9_%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) значення умови вказаної програмістом.

Найбільш часто умовний перехід має дві стадії: на першій відбувається порівняння між собою деяких величин, що визначають умову переходу, а на другій виконується сам перехід.

Необхідність коректної обробки умовних переходів накладає серйозний відбиток на логіку роботи сучасних конвеєрних процесорів[[1]](https://uk.wikipedia.org/wiki/%D0%A3%D0%BC%D0%BE%D0%B2%D0%BD%D0%B8%D0%B9_%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D1%96%D0%B4#cite_note-1). Умовні переходи можуть виконуватися двома способами. Виконувані переходи часто змінюють значення лічильника команд [процесора](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80) на обчислене значення адреси переходу. Невиконувані — додають до значення лічильника команд число, рівне довжині поточної команди в байтах, для переходу до виконання наступної команди. Неправильне визначення типу умовного переходу може призводити до виникнення суттєвих затримок в роботі конвеєра і відповідно до великої втрати продуктивності [комп'ютера](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80).

**Цикл** ([лат.](https://uk.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B8%D0%BD%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *cyclus*, від [грец.](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B5%D1%86%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) kuklos — [круг](https://uk.wikipedia.org/wiki/%D0%9A%D1%80%D1%83%D0%B3)) — завершена [послідовність](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%81%D0%BB%D1%96%D0%B4%D0%BE%D0%B2%D0%BD%D1%96%D1%81%D1%82%D1%8C) функціонування матеріально-технічних компонентів об'єкта ([явища](https://uk.wikipedia.org/wiki/%D0%AF%D0%B2%D0%B8%D1%89%D0%B5)) в просторі та часі. Коло подій, явищ, частин.

Цикл — сукупність взаємозв'язаних явищ, процесів, що створюють закінчене коло розвитку протягом якогось проміжку часу (напр., Ц. виробничий, Ц. парової машини, двигуна внутрішнього згоряння). Замкнений Ц. — технологічний процес, за яким певний продукт переробки багаторазово повертається до вихідної операції і бере участь у наступному циклі переробки. У гірничій справі це, напр., процес подрібнення руди в замкненому Ц., багаторазове використання технологічної води в Ц. збагачення після її реґенерації (прояснення).

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Запустити 2вимірний цикл в якому будуть перебиратись елементи 2 масивів на знаходження одинакових елементів

## Висновок

Я навчився працювати з 2вимірними циклами

## Додаток(а)

#include <iostream>

using namespace std;

float A[8], B[8], k;

int main()

{

std::cout << "Vvedit masiv A[8]:" << std::endl;

for (int i=0; i<8; i++){

std::cin >> A[i];

}

std::cout << "Vvedit masiv B[8]:" << std::endl;

for (int i=0; i<8; i++){

std::cin >> B[i];

}

std::cout << "4isla: " << std::ends;

for (int i=0; i<8; i++){

for (int j=0; j<8; j++){

if(A[i] == B[j]){

k++;

std::cout <<A[i] << " " << std::ends;

}

}

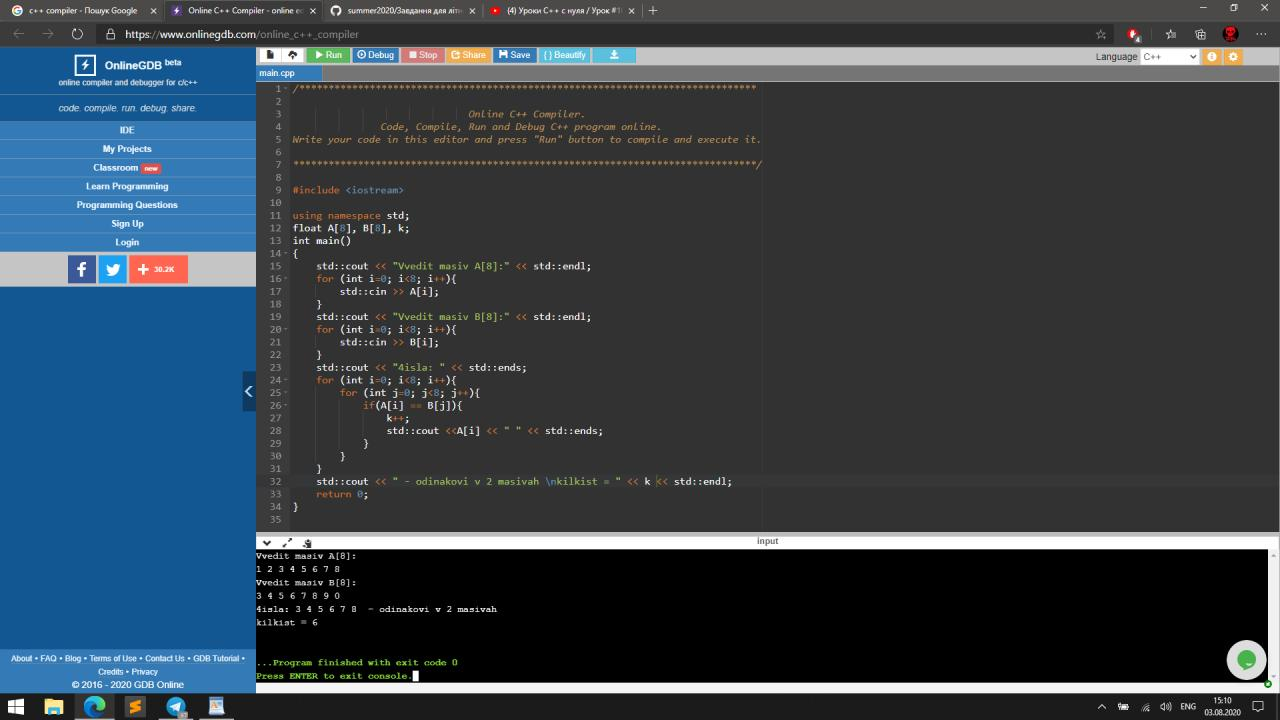
}

std::cout << " - odinakovi v 2 masivah \nkilkist = " << k << std::endl;

return 0;

}

## Додаток(б)



# Завдання 4

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

15. Визначити кількість ненульових елементів в матриці G(6,6). Вивести матрицю та кількість ненульових елементів

## ТЕОРЕТИЧНІ ВІДОМОСТІ

**Цикл** ([лат.](https://uk.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B8%D0%BD%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *cyclus*, від [грец.](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B5%D1%86%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) kuklos — [круг](https://uk.wikipedia.org/wiki/%D0%9A%D1%80%D1%83%D0%B3)) — завершена [послідовність](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%81%D0%BB%D1%96%D0%B4%D0%BE%D0%B2%D0%BD%D1%96%D1%81%D1%82%D1%8C) функціонування матеріально-технічних компонентів об'єкта ([явища](https://uk.wikipedia.org/wiki/%D0%AF%D0%B2%D0%B8%D1%89%D0%B5)) в просторі та часі. Коло подій, явищ, частин.

Цикл — сукупність взаємозв'язаних явищ, процесів, що створюють закінчене коло розвитку протягом якогось проміжку часу (напр., Ц. виробничий, Ц. парової машини, двигуна внутрішнього згоряння). Замкнений Ц. — технологічний процес, за яким певний продукт переробки багаторазово повертається до вихідної операції і бере участь у наступному циклі переробки. У гірничій справі це, напр., процес подрібнення руди в замкненому Ц., багаторазове використання технологічної води в Ц. збагачення після її реґенерації (прояснення).

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Створення підпрограми для обрахування кількості значеннь в матриці які не дорівнюють 0

## Висновок

Я навчився працювати з двовимірними масивами та матрицями

## Додаток(а)

# include<stdio.h>

#include <stdlib.h>

#include <time.h>

#define N 6

void Zap(int G[][N], int n)

{

int i, j;

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

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

G[i][j] = rand()%10-1;

}

void Vuvid(int G[][N], int n)

{

int i, j;

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

{

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

printf("%d ", G[i][j]);

printf("\n");

}

}

void Kil(int G[][N], int n)

{int i,j,k=0;

for(i=0; i<6; i++)

for(j=0; j<6; j++)

if(G[i][j] !=0 )

k++;

printf("\nK= = %d", k);

}

int main()

{

int G[N][N];

srand(time(NULL));

Zap(G, N);

printf("\nMatrix:");

printf("\n");

Vuvid(G, N);

Kil(G,N);

return 0;}

## Додаток(б)

# Завдання 5

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

15. Визначити мiнiмальнi елементи в кожному рядку заданої матриці T(8,3). Вивести мiнiмальнi елементи та номери відповідних рядків.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

**Умовний перехід** — конструкція [мови програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), яка дозволяє виконувати різні дії залежно від [булевого](https://uk.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D1%96%D1%87%D0%BD%D0%B8%D0%B9_%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) значення умови вказаної програмістом.

Найбільш часто умовний перехід має дві стадії: на першій відбувається порівняння між собою деяких величин, що визначають умову переходу, а на другій виконується сам перехід.

Необхідність коректної обробки умовних переходів накладає серйозний відбиток на логіку роботи сучасних конвеєрних процесорів[[1]](https://uk.wikipedia.org/wiki/%D0%A3%D0%BC%D0%BE%D0%B2%D0%BD%D0%B8%D0%B9_%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D1%96%D0%B4#cite_note-1). Умовні переходи можуть виконуватися двома способами. Виконувані переходи часто змінюють значення лічильника команд [процесора](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80) на обчислене значення адреси переходу. Невиконувані — додають до значення лічильника команд число, рівне довжині поточної команди в байтах, для переходу до виконання наступної команди. Неправильне визначення типу умовного переходу може призводити до виникнення суттєвих затримок в роботі конвеєра і відповідно до великої втрати продуктивності [комп'ютера](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80).

**Цикл** ([лат.](https://uk.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B8%D0%BD%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *cyclus*, від [грец.](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B5%D1%86%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) kuklos — [круг](https://uk.wikipedia.org/wiki/%D0%9A%D1%80%D1%83%D0%B3)) — завершена [послідовність](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%81%D0%BB%D1%96%D0%B4%D0%BE%D0%B2%D0%BD%D1%96%D1%81%D1%82%D1%8C) функціонування матеріально-технічних компонентів об'єкта ([явища](https://uk.wikipedia.org/wiki/%D0%AF%D0%B2%D0%B8%D1%89%D0%B5)) в просторі та часі. Коло подій, явищ, частин.

Цикл — сукупність взаємозв'язаних явищ, процесів, що створюють закінчене коло розвитку протягом якогось проміжку часу (напр., Ц. виробничий, Ц. парової машини, двигуна внутрішнього згоряння). Замкнений Ц. — технологічний процес, за яким певний продукт переробки багаторазово повертається до вихідної операції і бере участь у наступному циклі переробки. У гірничій справі це, напр., процес подрібнення руди в замкненому Ц., багаторазове використання технологічної води в Ц. збагачення після її реґенерації (прояснення).

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Знаходження мінімального елементу в рядку шляхом зациклення процедури з першим елементом нового рядка

## Висновок

Я навчився використовувати цикли на знаходити мінімальний елемент

## Додаток(а)

#include <iostream>

using namespace std;

float T[8][3];

float minimal;

int i, j;

int main()

{

std::cout << "Vvedit matriciy rozmirom 8 stovbciv po 3 zna4enia" << std::endl;

for (i=0; i<8; i++){

for (j=0; j<3; j++){

std::cin >> T[i][j];

}

}

for (i=0; i<8; i++){

minimal = T[i][0];

for (j=0; j<3; j++){

if (minimal > T[i][j]){

minimal = T[i][j];

}

}

std::cout << "V riadky №"<< i << " minimalne zna4ienia = " << minimal << std::endl;

}

return 0;

}

## Додаток(б)

# Завдання 6

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

15. Задано матрицю К(7,7). Серед елементів, розташованих нижче побічної діагоналі, визначити добуток ненульових елементів. Вивести вихідну матрицю, максимальний і мінімальний елементи, їх індекси та обчислений добуток.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

**Умовний перехід** — конструкція [мови програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), яка дозволяє виконувати різні дії залежно від [булевого](https://uk.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D1%96%D1%87%D0%BD%D0%B8%D0%B9_%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) значення умови вказаної програмістом.

Найбільш часто умовний перехід має дві стадії: на першій відбувається порівняння між собою деяких величин, що визначають умову переходу, а на другій виконується сам перехід.

Необхідність коректної обробки умовних переходів накладає серйозний відбиток на логіку роботи сучасних конвеєрних процесорів[[1]](https://uk.wikipedia.org/wiki/%D0%A3%D0%BC%D0%BE%D0%B2%D0%BD%D0%B8%D0%B9_%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D1%96%D0%B4#cite_note-1). Умовні переходи можуть виконуватися двома способами. Виконувані переходи часто змінюють значення лічильника команд [процесора](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80) на обчислене значення адреси переходу. Невиконувані — додають до значення лічильника команд число, рівне довжині поточної команди в байтах, для переходу до виконання наступної команди. Неправильне визначення типу умовного переходу може призводити до виникнення суттєвих затримок в роботі конвеєра і відповідно до великої втрати продуктивності [комп'ютера](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80).

**Цикл** ([лат.](https://uk.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B8%D0%BD%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *cyclus*, від [грец.](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B5%D1%86%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) kuklos — [круг](https://uk.wikipedia.org/wiki/%D0%9A%D1%80%D1%83%D0%B3)) — завершена [послідовність](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%81%D0%BB%D1%96%D0%B4%D0%BE%D0%B2%D0%BD%D1%96%D1%81%D1%82%D1%8C) функціонування матеріально-технічних компонентів об'єкта ([явища](https://uk.wikipedia.org/wiki/%D0%AF%D0%B2%D0%B8%D1%89%D0%B5)) в просторі та часі. Коло подій, явищ, частин.

Цикл — сукупність взаємозв'язаних явищ, процесів, що створюють закінчене коло розвитку протягом якогось проміжку часу (напр., Ц. виробничий, Ц. парової машини, двигуна внутрішнього згоряння). Замкнений Ц. — технологічний процес, за яким певний продукт переробки багаторазово повертається до вихідної операції і бере участь у наступному циклі переробки. У гірничій справі це, напр., процес подрібнення руди в замкненому Ц., багаторазове використання технологічної води в Ц. збагачення після її реґенерації (прояснення).

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

В 2 вимірному циклі запустити пошук та накопичення добутку не нульових елементів за умови знаходження цього елементу нижче бічної діагоналі

## Висновок

Я навчився розбивати матрицю по бічній діагоналі знаходити добуток ненульових елементів та шукати індекс елемента

## Додаток(а)

#include <iostream>

using namespace std;

float K[7][7];

float minimal, dobutok = 1.0;

int i, j;

int main()

{

std::cout << "Vvedit matriciy rozmirom 7 stovbciv po 7 zna4en`" << std::endl;

for (i=0; i<7; i++){

for (j=0; j<7; j++){

std::cin >> K[i][j];

}

}

minimal = K[0][0];

for (i=0; i<7; i++){

for (j=i; j>=0; j--){

if (K[i][j] != 0){

dobutok \*= K[i][j];

}

if (minimal > K[i][j]){

minimal = K[i][j];

}

}

}

std::cout << "Dobutok = "<< dobutok << "\nminimalne zna4ienia = " << minimal << "\nminimalni zna4ienia mayt` indexi:" << std::ends;

for (i=0; i<7; i++){

for (j=i; j>=0; j--){

if (K[i][j] == minimal){

std::cout << " [" << i << "][" << j << "], " << std::ends;

}

}

}

return 0;

}

## Додаток(б)

# Завдання 7

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

15. Дано цілочислову матрицю *А* розмірності *n*x*m.* Поміняти місцями ліву і праву половини масиву.

*Вхідні дані:* *n =* 3; *m =* 4; .

*Вихідні дані:* .

## ТЕОРЕТИЧНІ ВІДОМОСТІ

**Умовний перехід** — конструкція [мови програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), яка дозволяє виконувати різні дії залежно від [булевого](https://uk.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D1%96%D1%87%D0%BD%D0%B8%D0%B9_%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) значення умови вказаної програмістом.

Найбільш часто умовний перехід має дві стадії: на першій відбувається порівняння між собою деяких величин, що визначають умову переходу, а на другій виконується сам перехід.

Необхідність коректної обробки умовних переходів накладає серйозний відбиток на логіку роботи сучасних конвеєрних процесорів[[1]](https://uk.wikipedia.org/wiki/%D0%A3%D0%BC%D0%BE%D0%B2%D0%BD%D0%B8%D0%B9_%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D1%96%D0%B4#cite_note-1). Умовні переходи можуть виконуватися двома способами. Виконувані переходи часто змінюють значення лічильника команд [процесора](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80) на обчислене значення адреси переходу. Невиконувані — додають до значення лічильника команд число, рівне довжині поточної команди в байтах, для переходу до виконання наступної команди. Неправильне визначення типу умовного переходу може призводити до виникнення суттєвих затримок в роботі конвеєра і відповідно до великої втрати продуктивності [комп'ютера](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80).

**Цикл** ([лат.](https://uk.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B8%D0%BD%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *cyclus*, від [грец.](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B5%D1%86%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) kuklos — [круг](https://uk.wikipedia.org/wiki/%D0%9A%D1%80%D1%83%D0%B3)) — завершена [послідовність](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%81%D0%BB%D1%96%D0%B4%D0%BE%D0%B2%D0%BD%D1%96%D1%81%D1%82%D1%8C) функціонування матеріально-технічних компонентів об'єкта ([явища](https://uk.wikipedia.org/wiki/%D0%AF%D0%B2%D0%B8%D1%89%D0%B5)) в просторі та часі. Коло подій, явищ, частин.

Цикл — сукупність взаємозв'язаних явищ, процесів, що створюють закінчене коло розвитку протягом якогось проміжку часу (напр., Ц. виробничий, Ц. парової машини, двигуна внутрішнього згоряння). Замкнений Ц. — технологічний процес, за яким певний продукт переробки багаторазово повертається до вихідної операції і бере участь у наступному циклі переробки. У гірничій справі це, напр., процес подрібнення руди в замкненому Ц., багаторазове використання технологічної води в Ц. збагачення після її реґенерації (прояснення).

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Замінити значення стовпчиків матриці один на інший за допомогою 3 змінної tmp

## Висновок

Я навчився заміщувати значення в матриці за допомогою 3 змінної

## Додаток(а)

#include <iostream>

using namespace std;

float minimal, dobutok = 1.0;

int i, j, n, m;

int main()

{

std::cout << "Vvedit rozmir matrici n: " << std::ends;

std::cin >> n;

std::cout << "Vvedit rozmir matrici m: " << std::ends;

std::cin >> m;

float A[n][m];

std::cout << "Vvedit matriciy rozmirom " << n << " stovbciv po " << m << " zna4en`" << std::endl;

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

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

std::cin >> A[i][j];

}

}

int tmp;

if (m % 2 == 0){

int tmp1 = m/2;

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

for (j=0; j<tmp1; j++){

tmp = A[i][j];

A[i][j] = A[i][j+tmp1];

A[i][j+tmp1] = tmp;

}

}

} else {

float tmp2 = m/2;

int tmp1 = tmp2;

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

for (j=0; j<tmp1; j++){

tmp = A[i][j];

A[i][j] = A[i][j+(tmp1+1)];

A[i][j+tmp1+1] = tmp;

}

}

}

std::cout << "Nova matriciy rozmirom " << n << " stovbciv po " << m << " zna4en` =" << std::endl;

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

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

std::cout << A[i][j] << " " << std::ends;

}

std::cout << std::endl;

}

return 0;

}

## Додаток(б)

# Завдання 8

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

1. Дано матрицю *А* розмірності *nxп*. Використовуючи вказівники знайти мінімальний елемент масиву, вивести його адресу та координати.

*Вхідні дані:* *n = 3*; .

*Вихідні дані:* *min =* 11, *адреса: 566358; i = 1, j = 1*.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

**Вказівни́к**[[1]](https://uk.wikipedia.org/wiki/%D0%92%D0%BA%D0%B0%D0%B7%D1%96%D0%B2%D0%BD%D0%B8%D0%BA#cite_note-1), **пока́жчик** або **пока́зник** , іноді також **посилання**[[2]](https://uk.wikipedia.org/wiki/%D0%92%D0%BA%D0%B0%D0%B7%D1%96%D0%B2%D0%BD%D0%B8%D0%BA#cite_note-2) ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *pointer* або *reference*) — тип даних в комп'ютерних [мовах програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), об'єкт програми, що містить адресу в пам'яті [комп'ютера](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80) іншого об'єкта.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

За допомогою вказівника на перший елемент масиву знайти вказівник на мінімальне значення в масиві та вивести його дані

## Висновок

Я навчився працювати з вказівниками та освоїв їхнє призначення.

## Додаток(а)

#include <iostream>

using namespace std;

float minimal, dobutok = 1.0;

int i, j, n, ii, jj;

float \*ptrminimal = new float;

int main()

{

std::cout << "Vvedit rozmir matrici n: " << std::ends;

std::cin >> n;

float A[n][n];

std::cout << "Vvedit matriciy rozmirom " << n << " stovbciv po " << n << " zna4en`" << std::endl;

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

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

std::cin >> A[i][j];

}

}

\*ptrminimal = A[0][0];

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

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

if(\*ptrminimal > A[i][j]){

\*ptrminimal = A[i][j];

ii = i;

jj = j;

}

}

}

std::cout << "Minimalne = " << \*ptrminimal << ", adress = " << &ptrminimal << " Nomer = [" << ii << "][" << jj << "]" << std::endl;

delete ptrminimal;

return 0;

}

## Додаток(б)

## 

# Завдання 9

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

Нехай шість населених пунктів позначені номерами від 1 до 6 (величина **k**), а п’ять кандидатів – номерами від 1 до 5 (величина **п**). Кількість голосів, набраних кандидатами у кожному пункті визначається рандомно. Вивести на екран таблицю результатів голосування, де у рядках є дані з населених пунктів, а у стовпцях - дані щодо конкретних кандидатів. Визначити і вивести значення величин з додаткового завдання. Створити одновимірний масив з шуканими даними.

1. У кого з-поміж другого, четвертого і п’ятого кандидатів найви­щий рейтинг?

## ТЕОРЕТИЧНІ ВІДОМОСТІ

**Вказівни́к**[[1]](https://uk.wikipedia.org/wiki/%D0%92%D0%BA%D0%B0%D0%B7%D1%96%D0%B2%D0%BD%D0%B8%D0%BA#cite_note-1), **пока́жчик** або **пока́зник** , іноді також **посилання**[[2]](https://uk.wikipedia.org/wiki/%D0%92%D0%BA%D0%B0%D0%B7%D1%96%D0%B2%D0%BD%D0%B8%D0%BA#cite_note-2) ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *pointer* або *reference*) — тип даних в комп'ютерних [мовах програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), об'єкт програми, що містить адресу в пам'яті [комп'ютера](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80) іншого об'єкта.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Створити рандомний динамічний двовимірний масив та знайти найвищі значення, що задовільняють умову

## Висновок

Я навчився працювати з динамічною пам’яттю та використовувати функції рандому

## Додаток(а)

#include <iostream>

using namespace std;

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

{

int minimal, k;

float \*\*ptrarray = new float\* [6];

for (int i = 0; i < 6; i++){

ptrarray[i] = new float [5];

}

for (int i = 0; i < 6; i++){

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

ptrarray[i][j] = rand() % 100 + 1;

}

}

for (int i = 0; i < 6; i++){

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

cout << ptrarray[i][j] << " " << ends;

}

cout << endl;

}

for (int i = 0; i < 6; i++){

if ((i == 2) or (i == 4) or (i == 5)){

minimal = ptrarray[i][0];

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

if (minimal < ptrarray[i][j]){

minimal = ptrarray[i][j];

k = j;

}

}

std::cout << "V riadky № " << i << " naibilwe golosiv: " << minimal << " mae " << k+1 << " kandidat" << std::endl;

}

}

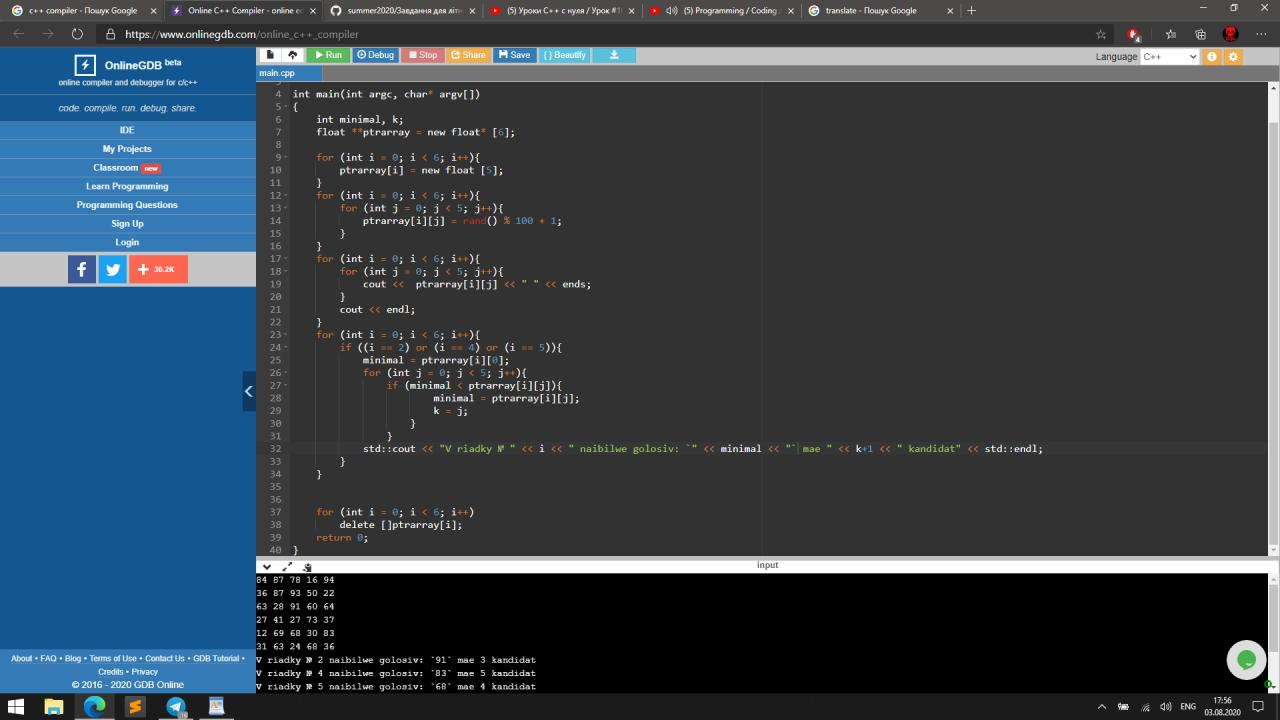
for (int i = 0; i < 6; i++)

delete []ptrarray[i];

return 0;

}

## Додаток(б)



# Завдання 10

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

Ввести прізвище, ім’я та по батькові як одне дане типу рядок. Визначити довжину рядка і кількість букв «а» у ньому. Виконати додатково завдання свого варіанта:

розглядаючи рядок як масив символів;

## ТЕОРЕТИЧНІ ВІДОМОСТІ

В мові програмування [C++](https://uk.wikipedia.org/wiki/C%2B%2B), [std](https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D0%B0_%D0%B1%D1%96%D0%B1%D0%BB%D1%96%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_C%2B%2B)::**string** є стандартним класом, який представляє текстовий [рядок](https://uk.wikipedia.org/wiki/%D0%A0%D1%8F%D0%B4%D0%BE%D0%BA_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)). Цей клас вирішує багато проблем, що виникають при роботі з [рядками Cі-стилю](https://uk.wikipedia.org/wiki/String.h), помістивши важку логіку управління пам'яттю на клас string, а не на програміста, і дозволили додавати NUL байт у рядок. Клас реалізує деякі типові операції з рядками такі як порівняння, конкатенація, пошук і заміна, і функцію для отримання [підрядків](https://uk.wikipedia.org/wiki/%D0%9F%D1%96%D0%B4%D1%80%D1%8F%D0%B4%D0%BE%D0%BA). Клас може бути побудований із рядка Сі-стилю, а рядок Cі-стилю може бути отримана з нього.

Рядки складаються з окремих символів C [char](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82), які як мінімум (і майже завжди) містять 8 бітів кожен. В сьогоднішньому застосуванні це часто не є "символом", а частиною [багатобайтового кодування символів](https://en.wikipedia.org/wiki/variable-width_encoding), такого як [UTF-8](https://uk.wikipedia.org/wiki/UTF-8).

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Зчитати рядок, порахувати кількість значень перевести в масив знайти кількість букв «а» та вивести ім’я в стовпчик

## Висновок

Я навчився працювати з типом даних string

## Додаток(а)

#include <iostream>

using namespace std;

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

{

string str;

int i, k = 0;

std::cout << "Vvedit im`ya: " << std::ends;

std::getline ( std::cin, str );

for (i=0; i<str.length(); i++){

if (str[i] == 'a'){

k++;

}

}

std::cout << "Kilkist bykv 'a' = " << k << ", vsogo bykv: " << str.length() << "\nIm`ya v stovp4ik:" << std::endl;

for (i=0; i<str.length(); i++){

if (str[i] == ' '){

k = i;

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

std::cout << str[j] << std::endl;

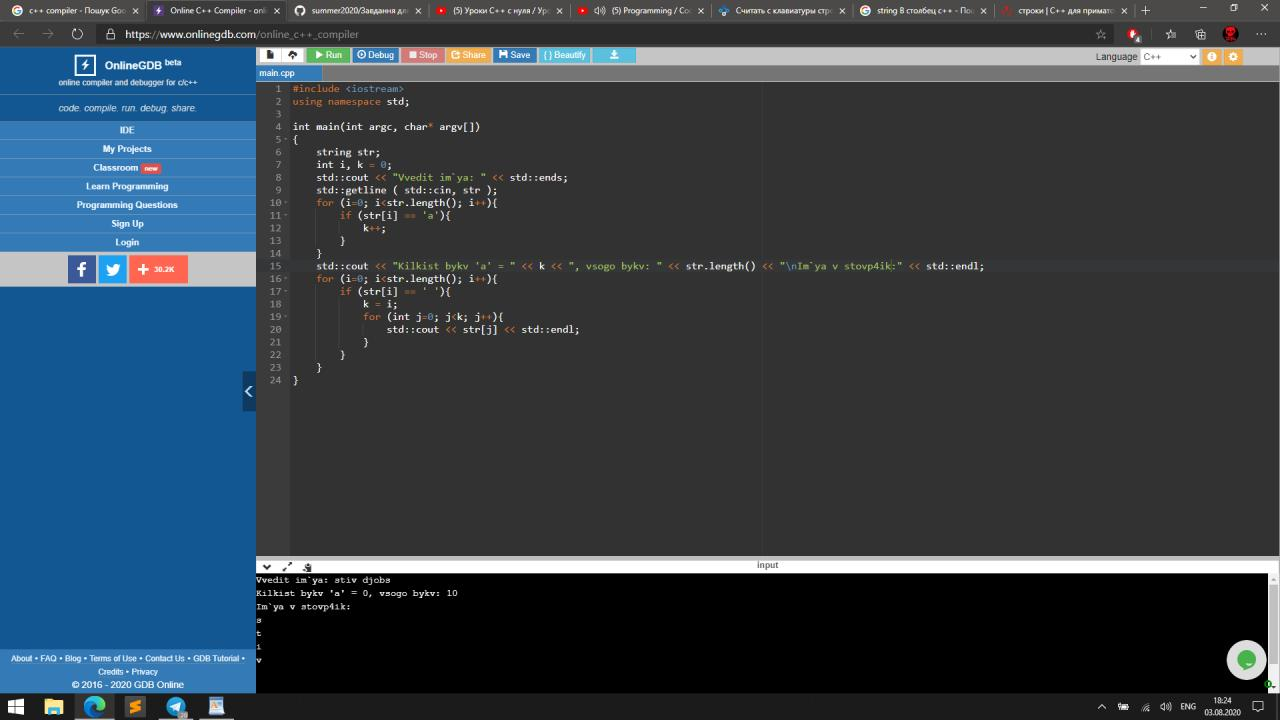
}

}

}

}

## Додаток(б)



# Завдання 11

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

15. Написати програму для визначення суми, яку потрібно сплатити за кредит на основі нарахувань: а) за простими відсотками (, де *Sn* – кінцева сума, *S0*– початкова сума, *і –* відсоткова ставка, *n* – термін вкладу); б) за складними відсотками (). При реалізації програми використати 2 функції.

*Вхідні дані:* *S0*= 1000 грн., *і =* 10%, *n* = 2 р*.*

*Вихідні дані:* а) *Sn =* 1200 грн; б) *Sn =* 1210 грн*.*

## ТЕОРЕТИЧНІ ВІДОМОСТІ

*Функція* — це іменована частина програми, до якої можна звертатися з її інших частин шляхом вказання імені, наприклад, SimpleFunc(5, 1.9). У даному випадку функція має ім’я SimpleFunc, якій передається два числових параметра.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Створити функцію для обчислення значення заданої формули

## Висновок

Я навчився використовувати функції в С++

## Додаток(а)

#include <iostream>

#include <math.h>

using namespace std;

float funct1(float sum, float proc, float year){

float result;

result = sum \* (1 + (proc/100) \* year);

return result;

}

float funct2(float sum, float proc, float year){

float result;

result = sum \* pow((1 + (proc/100)), year);

return result;

}

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

{

float S, procenti, years;

std::cout << "Vvedit po4atkovy symy kreduty " << std::ends;

std::cin >> S;

std::cout << "Vvedit procenti: " << std::ends;

std::cin >> procenti;

std::cout << "Vvedit strok: " << std::ends;

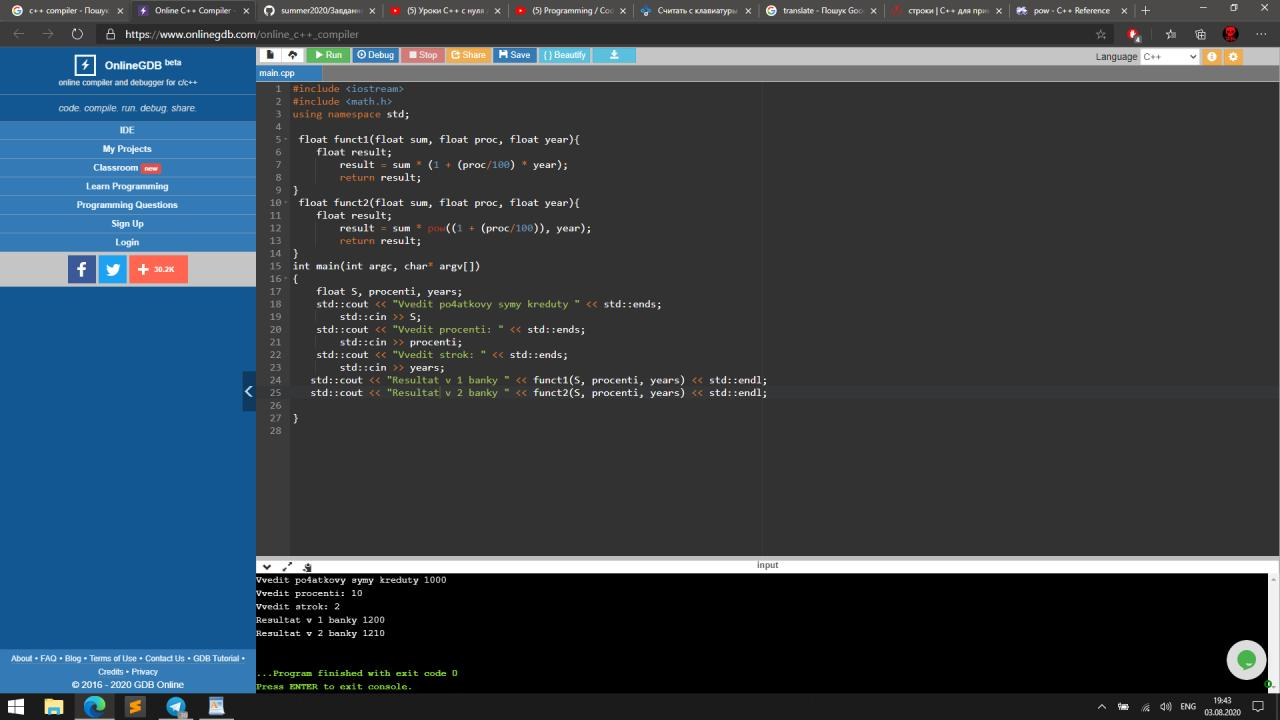
std::cin >> years;

std::cout << "Resultat v 1 banky " << funct1(S, procenti, years) << std::endl;

std::cout << "Resultat v 2 banky " << funct2(S, procenti, years) << std::endl;

}

## Додаток(б)



# Завдання 12

## ПОСТАНОВКА ЗАДАЧІ

### Опис завдання

15. В магазині дитячих товарів є відомості про наявні в продажу іграшки:

– назва іграшки (наприклад, лялька, м’яч, конструктор і т.п.);

– адреса заводу-виготовника (місто, вулиця, номер будинку);

– вартість іграшки (в гривнях);

– вікова категорія, на яку орієнтована дана іграшка (наприклад, від 2 до 5 років);

– чи може іграшка використовуватися для спортивних ігор (так чи ні).

Серед іграшок з Києва знайти найдорожчу (її назву і вартість) і найдешевшу (її назву і вартість). Для цього визначити функцію, яка повертає структуру, поля якої містять необхідну інформацію. Серед іграшок, призначених для спортивних ігор, визначити ту, яка орієнтована на наймолодшу вікову категорію (з наявних).

## ТЕОРЕТИЧНІ ВІДОМОСТІ

В мові C основним способом організації даних були структури. Структура складається з набору полів, які ніяк не захищені. Якщо елементи структури мають змінну довжину, їх представляють у вигляді вказівників. Виділення і звільнення пам'яті під ці вказівники робляться вручну. Так, наприклад, одновимірний масив змінної довжини в мові C з перевіркою меж може представлятися таким чином:

**struct** Array {

double\* val;

int len;

};

void FreeArray (**const** **struct** Array\*);

void AllocArray (**const** **struct** Array\*, int len);

double Elem (**const** **struct** Array\*, int i);

void ChangeElem (**const** **struct** Array\*, int i, double x);

Така реалізація небезпечна і неефективна з багатьох причин:

* Необхідно викликати FreeArray і AllocArray. Програміст може забути викликати одну з цих функцій, або викликати її дуже рано/запізно, або двічі, або з вказівником на неправильний масив. Все це приводить до помилок, що важко виявити.
* Функції Elem і ChangeElem повільні.
* Немає ніякого способу перешкодити програмістам створювати і інші функції для роботи із структурою Array. Ці функції можуть робити з полями len і val будь-що.
* Немає ніякого способу перешкодити програмістам безпосередньо міняти поля len і val.
* Присвоєння об'єктів типу struct Array приведе до того, що їхні поля val указуватимуть на одну і ту ж область пам'яті. Немає ніякого способу ні заборонити присвоєння, ні змінити таку поведінку.

Мова Сі++, використовуючи ООП, усуває всі ці проблеми.

## СХЕМИ АЛГОРИТМІВ ОСНОВНИХ ФУНКЦІЙ

Створити структуру за якою буде формулюватися масив даних, створити функції введення даних, пошуку та виведення

## Висновок

Я навчився використовувати структури в С++ та передавати масив структур в функцію

## Додаток(а)

#include <iostream>

#include <sstream>

#include <string>

using namespace std;

struct Toys{

string name;

string adress;

int price;

int ages;

bool games;

};

void f1(struct Toys\* toy, int n){

int prices = toy[0].price, j;

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

if ((toy[i].adress == "Kiiv") && (prices <= toy[i].price)){

prices = toy[i].price;

j = i;

}

}

std::cout<< "Naidoroj4a igrawka: " << toy[j].name << " z cinoiy: " << toy[j].price <<endl;

}

void f2(struct Toys\* toy, int n){

int prices2 = toy[0].price, j2;

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

if ((toy[i].adress == "Kiiv") && (prices2 >= toy[i].price)){

prices2 = toy[i].price;

j2 = i;

}

}

std::cout<< "Naidewevwa igrawka: " << toy[j2].name << " z cinoiy: " << toy[j2].price <<endl;

}

void f3(struct Toys\* toy, int n){

int age = toy[0].ages, j;

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

if ((toy[i].games == true) && (age > toy[i].ages)){

age = toy[i].ages;

j = i;

}

}

std::cout<< "Naimolodwa igrawka: " << toy[j].name << " z cinoiy: " << toy[j].price << " ta vikom vid: " << toy[j].ages <<endl;

}

void Vvid(struct Toys\* toy, int n){

string name, adress;

int price, age, game;

std::cout << "\n Igrawka №" << n-2 << std::endl;

std::cout << "Vvedit nazvy igrawki: " << std::ends;

std::cin >> name;

std::cout << "Vvedit adress igrawki: " << std::ends;

std::cin >> adress;

std::cout << "Vvedit price igrawki: " << std::ends;

std::cin >> price;

std::cout << "Vvedit minimalnii vik dutunu: " << std::ends;

std::cin >> age;

std::cout << "Sportivna igrawka? (1 - tak / 0 - ni) " << std::ends;

std::cin >> game;

toy[n].name = name;

toy[n].adress = adress;

toy[n].price = price;

toy[n].ages = age;

toy[n].games = game;

}

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

{

string str;

int n, i;

std::cout << "Vvedit kilkist igrawok: " << std::ends;

std::cin >> n;

Toys toy[n+4] = { {"Ball", "Kiiv", 100, 10, true}, {"Doll", "Kiiv", 1002, 3, false}, {"Circle", "Kiiv", 352, 5, true}};

for (i=3; i<n+3; i++){

Vvid(toy, i);

}

std::cout << "Results: " << std::endl;

f1(toy, n+4);

f2(toy, n+4);

f3(toy, n+4);

return 0;

}

## Додаток(б)

