# Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 1 з дисципліни

«Основи програмування-1.

Базові конструкції»

«Обчислення арифметичних виразів»

Варіант 28

Виконав студент ІП-11 Сідак Кирил Ігорович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 2021

**Варіант 28**

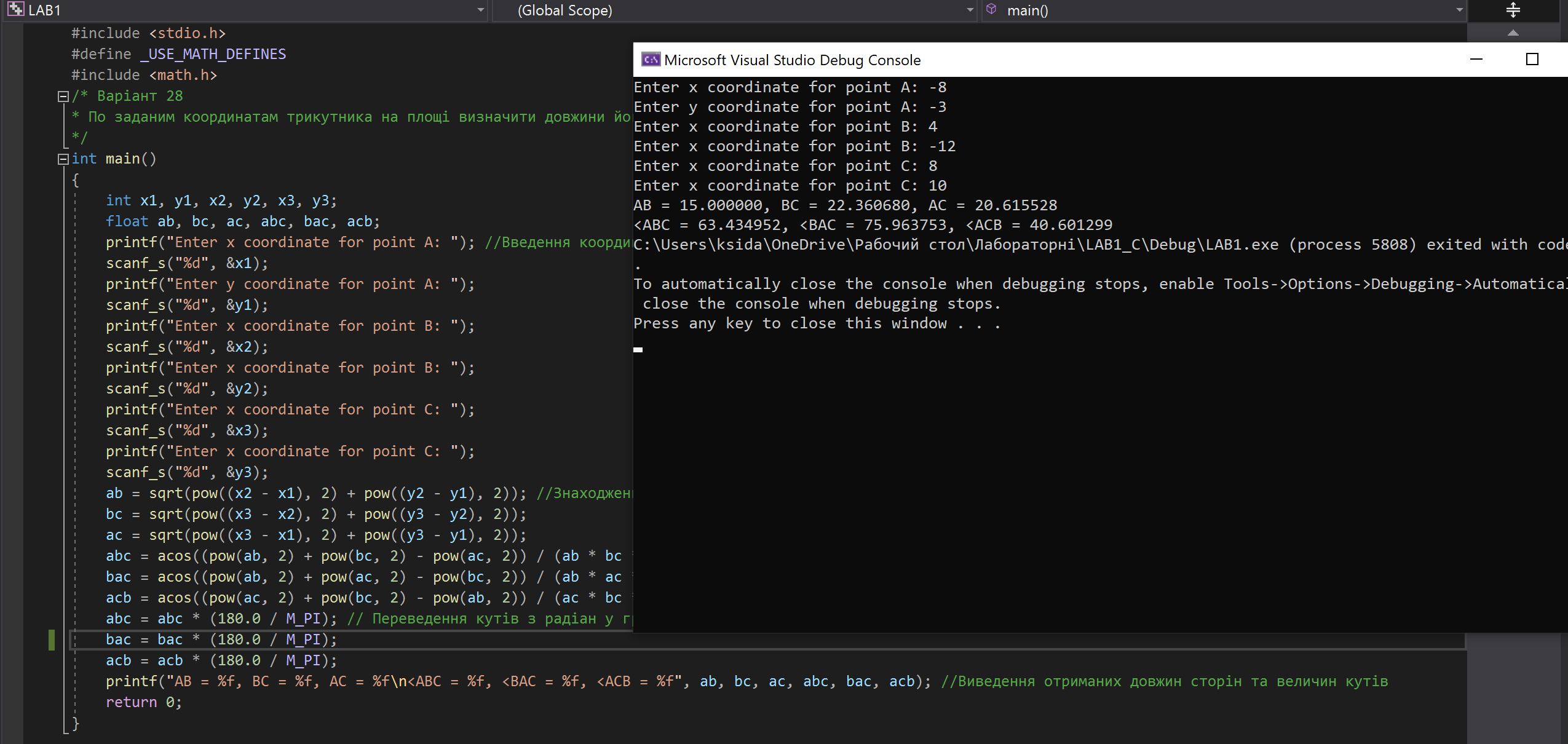
По заданим координатам трикутника на площі визначити довжини його сторін та величини кутів.

**Постановка задачі:** по заданим координатам вершин трикутника обчислити довжини сторін трикутника, а, використовуючи сторони, обчислити величини кутів трикутника.

**Програма на C:**

|  |
| --- |
| #include <stdio.h> |
|  | #define \_USE\_MATH\_DEFINES |
|  | #include <math.h> |
|  | /\* Варіант 28 |
|  | \* По заданим координатам трикутника на площі визначити довжини його сторін і величини кутів. |
|  | \*/ |
|  | int main() |
|  | { |
|  | int x1, y1, x2, y2, x3, y3; |
|  | float ab, bc, ac, abc, bac, acb; |
|  | printf("Enter x coordinate for point A: "); //Введення координат вершин трикутника |
|  | scanf\_s("%d", &x1); |
|  | printf("Enter y coordinate for point A: "); |
|  | scanf\_s("%d", &y1); |
|  | printf("Enter x coordinate for point B: "); |
|  | scanf\_s("%d", &x2); |
|  | printf("Enter x coordinate for point B: "); |
|  | scanf\_s("%d", &y2); |
|  | printf("Enter x coordinate for point C: "); |
|  | scanf\_s("%d", &x3); |
|  | printf("Enter x coordinate for point C: "); |
|  | scanf\_s("%d", &y3); |
|  | ab = sqrt(pow((x2 - x1), 2) + pow((y2 - y1), 2)); //Знаходження довжин сторін |
|  | bc = sqrt(pow((x3 - x2), 2) + pow((y3 - y2), 2)); |
|  | ac = sqrt(pow((x3 - x1), 2) + pow((y3 - y1), 2)); |
|  | abc = acos((pow(ab, 2) + pow(bc, 2) - pow(ac, 2)) / (ab \* bc \* 2)); //Знаходження величин кутів |
|  | bac = acos((pow(ab, 2) + pow(ac, 2) - pow(bc, 2)) / (ab \* ac \* 2)); |
|  | acb = acos((pow(ac, 2) + pow(bc, 2) - pow(ab, 2)) / (ac \* bc \* 2)); |
|  | abc = abc \* (180.0 / M\_PI); // Переведення кутів з радіан у градуси |
|  | bac = bac \* (180.0 / M\_PI); |
|  | acb = acb \* (180.0 / M\_PI); |
|  | printf("AB = %f, BC = %f, AC = %f\n<ABC = %f, <BAC = %f, <ACB = %f", ab, bc, ac, abc, bac, acb); //Виведення отриманих довжин сторін та величин кутів |
|  | return 0; |
|  | } |

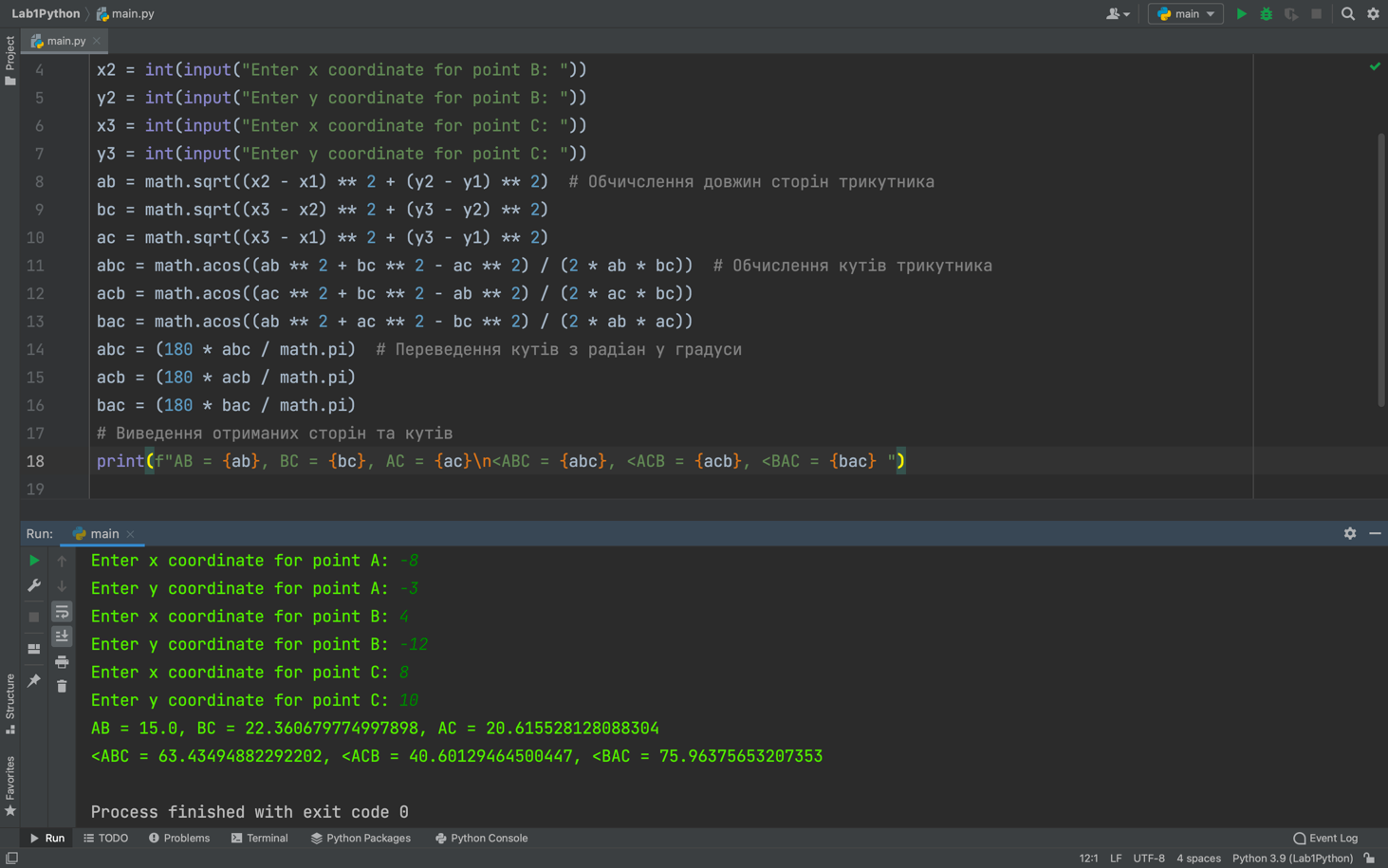
**Результат на C:**



**Програма на Python:**

|  |  |
| --- | --- |
|  | import math |
|  | x1 = int(input("Enter x coordinate for point A: ")) # Введення координат користувачем |
|  | y1 = int(input("Enter y coordinate for point A: ")) |
|  | x2 = int(input("Enter x coordinate for point B: ")) |
|  | y2 = int(input("Enter y coordinate for point B: ")) |
|  | x3 = int(input("Enter x coordinate for point C: ")) |
|  | y3 = int(input("Enter y coordinate for point C: ")) |
|  | ab = math.sqrt((x2 - x1) \*\* 2 + (y2 - y1) \*\* 2) # Обчичслення довжин сторін трикутника |
|  | bc = math.sqrt((x3 - x2) \*\* 2 + (y3 - y2) \*\* 2) |
|  | ac = math.sqrt((x3 - x1) \*\* 2 + (y3 - y1) \*\* 2) |
|  | abc = math.acos((ab \*\* 2 + bc \*\* 2 - ac \*\* 2) / (2 \* ab \* bc)) # Обчислення кутів трикутника |
|  | acb = math.acos((ac \*\* 2 + bc \*\* 2 - ab \*\* 2) / (2 \* ac \* bc)) |
|  | bac = math.acos((ab \*\* 2 + ac \*\* 2 - bc \*\* 2) / (2 \* ab \* ac)) |
|  | abc = (180 \* abc / math.pi) # Переведення кутів з радіан у градуси |
|  | acb = (180 \* acb / math.pi) |
|  | bac = (180 \* bac / math.pi) |
|  | # Виведення отриманих сторін та кутів |
|  | print(f"AB = {ab}, BC = {bc}, AC = {ac}\n<ABC = {abc}, <ACB = {acb}, <BAC = {bac} ") |

**Результат на Python:**



**Висновок:** Отже, використовуючи формулу відстані між двома точками, отримаємо коректний результат для довжин сторін трикутника, а використовуючи теорему косинусів, отримаємо коректний результат для величин кутів у програмі.