

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 №6 з дисципліни
«Основи програмування»
«Організація підпрограм»
Варіант 34

Виконав студент ІП-1134 Шамков Іван Дмитрович
(прізвище, ім'я, по батькові)

Перевірив викладач Вітковська Ірина Іванівна
(прізвище, ім'я, по батькові)

Київ 2021
Лабораторна робота №5
Організація підпрограм

Лабораторна робота 6

Організація підпрограм

Мета – набути навичок складання і використання підпрограм користувача.

Варіант: 34

Умова задачі:

34. Нехай n трикутників задані двома сторонами і кутом між ними.
Визначити трикутник, що має максимальну площу.

Математична модель:

Змінна	Тип	Ім'я	Призначення
Кількість трикутників, що треба вивести	Цілий	N	Початкове дане
Сторона А	Дійсний	side_a	Початкове дане
Сторона В	Дійсний	side_b	Початкове дане
Кут між ними	Дійсний	angle	Початкове дане
Число π	Дійсний, константа	PI	Початкове дане
Номер поточної найбільшої змінної	Цілий	index	Проміжне значення
Початкова кількість трикутників	Цілий	number	Початкове дане
Площа трикутника	Дійсний	square	Проміжне значення
Максимальна площа	Дійсний	max_square	Результат

Постановка задачі:

Отже, математичне формулювання нашої задачі полягає в тому, щоб отримати значення n , яке є кількістю трикутників. n разів ми питаємо у користувача потрібні параметри для обчислення площі трикутника. Перевіряємо розмір поточної площі із останньою

максимальною площею, що записали. Якщо вона більше, то записуємо нове значення.

Текст файла проекту:

Python:

```
from math import sin, pi #імпортуємо з бібліотеки функцію синус та константу PI

def deg_to_rad(angle): #функція, що переводить градуси в радіани
    angle=angle*(pi/180)
    return angle

def max_triangle(n:int, number, max_square=0, index=0): #функція для обрахунку площі заданих трикутників та пошуку найбільшої серед них
    if n==0:#Крайній випадок: обраховано n площ трикутників
        print("Трикутник під номером", number+1-index, "має максимальну площу", max_square)
        return

    #Запитуємо довжини сторін трикутника та значення кута між ними
    side_a=float(input("Enter length of side_a "))
    side_b=float(input("Enter length of side_b "))
    angle=float(input("Enter angle "))
    #обчислюємо площу трикутника
    square=0.5*side_a*side_b*sin(deg_to_rad(angle))

    if square>max_square:#Перевіряємо, чи поточна максимальна площа менша за площу тільки-но знайденого трикутника
        max_square=square#Якщо так, то записуємо нове значення максимальної площі
        index=n#Запам'ятовуємо номер трикутника з максимальною площею
    max_triangle(n-1, number, max_square, index)#Викликаємо цю ж функцію, в яку записуємо нові дані

n=int(input("Enter number of triangles"))#Запитуємо в користувача кількість трикутників
max_triangle(n, n)#Викликаємо функцію для пошуку найбільшої площі трикутника
```

C++:

```
#include <iostream>
#define _USE_MATH_DEFINES //для константи M_PI
#include <math.h>
using namespace std;

double deg_to_rad(double angle); //функція, що переводить градуси в радіани

void max_triangle(int n, int number, double max_square = 0, int index = 0); //функція для обрахунку площі заданих трикутників та пошуку найбільшої серед них

int main() {
    int n;
    //Запитуємо в користувача кількість трикутників
    cout << "Enter number of triangles" << endl;
    cin >> n;
    //Викликаємо функцію для пошуку найбільшої площі трикутника
    max_triangle(n, n);
    return 0;
}

double deg_to_rad(double angle){ ... }

void max_triangle(int n, int number, double max_square, int index )
{
    double side_a, side_b, angle;
    if (n == 0)
    {
        cout << "triangle with the number " << number + 1 - index << " has the biggest square and it is equal to: " << max_square;
        return;
    }
    //Запитуємо довжини сторін трикутника та значення кута між ними
    cout << "Enter side_a" << endl;
    cin >> side_a;
    cout << "Enter side_b" << endl;
    cin >> side_b;
    cout << "Enter angle" << endl;
    cin >> angle;
    //обчислюємо площу трикутника
    double square = 0.5 * side_a * side_b * sin(deg_to_rad(angle));
    //Перевіряємо, чи поточна максимальна площа менша за площу тільки-но знайденого трикутника
    if (square > max_square)
    {
        max_square = square;
        index = n;
    }
    //Викликаємо цю ж функцію, в яку записуємо нові дані
    max_triangle(n - 1, number, max_square, index);
}
```

Копії екранних форм:

Python:

```
Enter number of triangles 5
Enter length of side_a 12
Enter length of side_b 6
Enter angle 67
Enter length of side_a 14
Enter length of side_b 15
Enter angle 21
Enter length of side_a 2
Enter length of side_b 5
Enter angle 13
Enter length of side_a 60
Enter length of side_b 1
Enter angle 30
Enter length of side_a 5
Enter length of side_b 5
Enter angle 45
Трикутник під номером 2 має максимальну площу 37.62863470225653
```

C++:

```
Enter number of triangles
5
Enter side_a
12
Enter side_b
6
Enter angle
67
Enter side_a
14
Enter side_b
15
Enter angle
21
Enter side_a
2
Enter side_b
5
Enter angle
13
Enter side_a
60
Enter side_b
1
Enter angle
30
Enter side_a
5
Enter side_b
5
Enter angle
45
triangle with the number 2 has the biggest square and it is equal to: 37.6286
```

Висновок

Отже, виконавши цю лабораторну роботу, ми навчилися використовувати підпрограми. Проекти, на мою думку, розроблені коректно, адже заплановані елементи працюють, а саме: отримання даних від користувача, виклик функції пошуку найбільшого значення площі та переведення градусів у радіани. Перша функція викликає саму себе до того моменту, поки не буде обрахована задана кількість площ трикутників. До того моменту ми у самій же функції викликаємо її, але вже задаємо нові параметри. При цьому ми постійно запам'ятовуємо найбільшу площу трикутника та його номер. У кінці виводимо їх.