Основи програмування

*Додаток 1*

Міністерство освіти і науки України

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

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації

і управління

Звіт

з лабораторної роботи № 1 з дисципліни

«Основи програмування-1

Базові конструкції»

«Дослідження лінійних алгоритмів»

Варіант     31

Виконав студент       ІП-02 Прощенко Антон Юрійович

(шифр, прізвище, ім'я, по батькові)

Перевірив              Лебідь Сергій Олександрович

( прізвище, ім'я, по батькові)

Київ 2020

# Лабораторна робота 1 Дослідження лінійних алгоритмів Варіант 31

**Мета** – дослідити лінійні програмні специфікації для подання перетворювальних операторів та

операторів суперпозиції, набути практичних навичок їх використання під час складання лінійних

програмних специфікацій.

1. Постановка задачі

Задано сторону квадрата. Потрібно знайти його діагональ, периметр та площу. За теоремою Піфагора,

діагональ можна знайти, помноживши сторону квадрата на корінь квадратний з двох. Периметр дорівнює

сумі всіх сторін, а площа – стороні в квадраті.

1. Текст програми

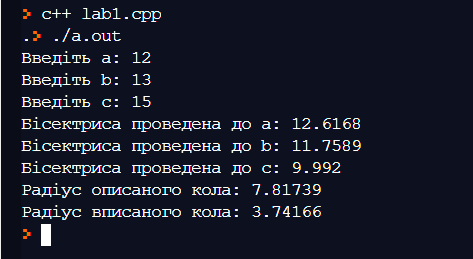
Python:

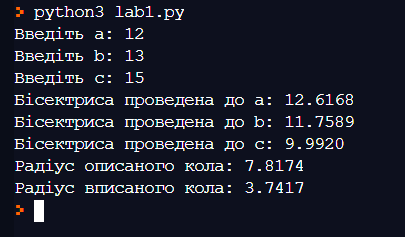
|  |
| --- |
| from math import sqrt  #input a = float(input("Введіть а: ")) b = float(input("Введіть b: ")) c = float(input("Введіть c: "))  def validate\_triangle(a, b, c):  if any(side <= 0 for side in [a,b,c]):  raise Exception("Вибачте, але сторони трикутника повинні бути більшими від нуля")   a, b, c = sorted([a, b, c])  if a + b <= c:  raise Exception("Вибачте, але трикутник існує тоді і тільки тоді, коли сума будь-яких двох його сторін більше третьою боку")  #a, b are sides of the angle bisector divides #side\_against\_bisector is the side bisector is drawn to def bisector(a, b, side\_against\_bisector):  c = side\_against\_bisector  return sqrt(a \* b \* (a + b + c) \* (a + b - c)) / (a + b)  validate\_triangle(a,b,c)  bisector\_a = bisector(c, b, a) print("Бісектриса проведена до а: {:.2f}".format(bisector\_a))  bisector\_b = bisector(a, c, b) print("Бісектриса проведена до b: {:.2f}".format(bisector\_b))  bisector\_c = bisector(a, b, c) print("Бісектриса проведена до c: {:.2f}".format(bisector\_c))  # evaluate radiuses # calculate the area  semiPerimeter = (a + b + c) / 2 area = sqrt(semiPerimeter\*(semiPerimeter-a)\*(semiPerimeter-b)\*(semiPerimeter-c))  R = (a \* b \* c) / (4 \* area) print("Радіус описаного кола: {:.2f}".format(R))  r = area / semiPerimeter print("Радіус вписаного кола: {:.2f}".format(r)) |

C++:

|  |
| --- |
| #include <iostream> #include <math.h> using namespace std;   float calcBisector(float a, float b,float sideAgainstBisector){  float c = sideAgainstBisector;  float result = sqrt(a \* b \* (a + b + c) \* (a + b - c)) / (a + b);  return result; }  void validateTriangle(float a, float b,float c){  if (a <= 0 || b <= 0 || c <= 0) {  throw "Вибачте, але сторони трикутника повинні бути більшими від нуля";  }  if (a + b <= c || a + c <= b || c + b <= a){  throw "Вибачте, але трикутник існує тоді і тільки тоді, коли сума будь-яких двох його сторін більше третьою боку";  } }  int main() {  float a;  float b;  float c;   cout << "Введіть а: ";  cin >> a;   cout << "Введіть b: ";  cin >> b;   cout << "Введіть c: ";  cin >> c;   float bisectorA = calcBisector(c, b, a);  float bisectorB = calcBisector(a, c, b);  float bisectorC = calcBisector(a, b, c);   cout << "Бісектриса проведена до a: " << bisectorA << "\n";  cout << "Бісектриса проведена до b: " << bisectorB << "\n";  cout << "Бісектриса проведена до c: " << bisectorC << "\n";    float semiPerimeter = (a + b + c) / 2;  float area = sqrt(semiPerimeter\*(semiPerimeter-a)\*(semiPerimeter-b)\*(semiPerimeter-c));   float R = (a \* b \* c) / (4 \* area);  cout << "Радіус описаного кола: " << R << "\n";   float r = area / semiPerimeter;  cout << "Радіус вписаного кола: " << r << "\n";   return 0; } |

1. Копії екранних форм результатів роботи





1. Висновок

Виконавши цю лабораторну роботу, я дослідив специфікації подання лінійних операторів та операторів суперпозиції. Успішно розв`язав задачу, склавши алгоритм. Набув практичних навичок використання псевдокоду, блок-схем під час складання лінійних алгоритмів.