

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 № 4 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Лінійні алгоритми»

Варіант 1

Виконав студент ІП-45 Янов Богдан Євгенійович

Перевірила старший викладач Вечерковська Анастасія Сергіївна

Київ 2024

Лабораторна робота №4

Тема - допоміжні алгоритми.

Мета – дослідити особливості роботи допоміжних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

Завдання

Реалізувати підпрограму для обчислення довжини відрізка АВ на координатній площині за координатами точок А(x_a , y_a) та В(x_b , y_b). Використати створену підпрограму в програмі для знаходження периметра трикутника, заданого координатами його вершин.

Постановка задач

1. Визначимо основні дії.
2. Деталізуємо дію обчислення довжини відрізка, як допоміжний алгоритм distance
3. Деталізуємо дію знаходження ab , bc , ca
4. Деталізуємо дію знаходження perimeter

Математична модель

Name	Type	Variable	Purpose
x_a , y_a , x_b , y_b , x_c , y_c	Real	Triangle vertices	Initial data
ab , bc , ca	Real	Triangle edges	Additional data
perimeter	Real	Triangle perimeter	Result

Псевдокод

Step 1

Begin

Input A(xa, ya)

Input B(xb, yb)

Input C(xc, yc)

Calculation of edges

Calculation of perimeter

Output perimeter

End

Step 2

Begin distance(xa, ya, xb, yb)

Return $\sqrt{\text{pow}(\text{xb} - \text{xa}, 2) + \text{pow}(\text{yb} - \text{ya}, 2)}$

End

Begin

Input A(xa, ya)

Input B(xb, yb)

Input C(xc, yc)

Calculation of edges

Calculation of perimeter

Output perimeter

End

Step 3

Begin distance(xa, ya, xb, yb)

Return $\sqrt{\text{pow}(\text{xb} - \text{xa}, 2) + \text{pow}(\text{yb} - \text{ya}, 2)}$

End

Begin

Input A(xa, ya)

Input B(xb, yb)

Input C(xc, yc)

ab = distance(xa, ya, xb, yb)

bc = distance(xb, yb, xc, yc)

ca = distance(xc, yc, xa, ya)

Calculation of perimeter

Output perimeter

End

Step 4

Begin distance(xa, ya, xb, yb)

Return sqrt(pow(xb - xa, 2), pow(yb - ya, 2))

End

Begin

Input A(xa, ya)

Input B(xb, yb)

Input C(xc, yc)

ab = distance(xa, ya, xb, yb)

bc = distance(xb, yb, xc, yc)

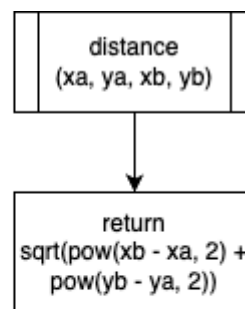
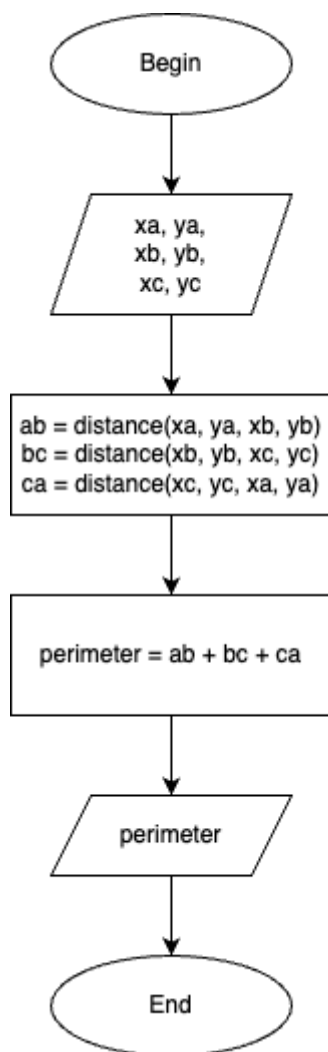
ca = distance(xc, yc, xa, ya)

perimeter = ab + bc + ca

Output perimeter

End

Блок-схема (крок 4)



Тестування алгоритму

A(xa, ya)	B(xb, yb)	C(xc, yc)	ab	bc	ca	perimeter
0, 0	0, 4	4, 0	4	5.65	4	13.65
1, 2	3, -1	5, 5	3.6	6.32	5	14.92

Код мовою C

```
#include <math.h>
#include <stdio.h>

double distance(double xa, double ya, double xb, double yb) {
    return sqrt(pow(xb - xa, 2) + pow(yb - ya, 2));
}

int main() {
    double xa, ya, xb, yb, xc, yc;

    printf("A (x, y): ");
    scanf("%lf %lf", &xa, &ya);

    printf("B (x, y): ");
    scanf("%lf %lf", &xb, &yb);

    printf("C (x, y): ");
    scanf("%lf %lf", &xc, &yc);

    double ab = distance(xa, ya, xb, yb);
    double bc = distance(xb, yb, xc, yc);
    double ca = distance(xc, yc, xa, ya);
```

```
double perimeter = ab + bc + ca;  
printf("P: %lf\n", perimeter);  
  
return 0;  
}
```

Висновок: Дослідив допоміжні алгоритми та набув практичних навичок їх використання.