

Soal 1. (30%)

Silahkan submit ADT Anda (*.c), kemudian upload pada slot yang disediakan (*.c)
Pada bagian header, Anda harus menggunakan
#include "point.h"
#include "boolean.h"
agar dapat digunakan untuk autograding.

Penamaan : P02_<NIM>_point.c

Soal 2. (35%)

```
function DalamRange (P, P1 : POINT, r : real) @ boolean
{ Mengembalikan true apabila P1 berada dalam jarak r dari P }
{ (jarak dari P1 ke P ≤ r) }
procedure GlideReflection (input/outputP : POINT, input sbX : boolean,
                           input delta : real)
{ I.S. P terdefinisi }
{ F.S. P dicerminkan terhadap sumbu x jika sbX bernilai true atau sumbu y jika
sbX bernilai false, kemudian digeser sebesar delta searah sumbu yang digunakan
sebagai sumbu pencerminan }
{ Contoh: jika P=<3,4>, maka GlideReflection(P,true,-2) menghasilkan P=<1,-4>.
Cerminkan terhadap sumbu-X menjadi <3,-4> kemudian digeser -2 satuan
arah sumbu-X menjadi <1,-4> }
```

Deskripsi

Tambahkan primitif-primitif diatas pada ADT point Anda. Buatlah sebuah program yang menerima input 2 buah point (**p** dan **p1**), sebuah nilai **r**, sebuah boolean **sbX** dan sebuah nilai **delta**.

Input

Baris pertama berisi nilai point **p** (x1, y1)

Baris kedua berisi nilai point **p1** (x2, y2)

Baris ketiga berisi sebuah nilai **r**.

Baris keempat berisi sebuah bilangan bulat [0, 1] yang mengartikan nilai boolean **sbX** (1 untuk true, 0 untuk false)

Baris kelima berisi nilai **delta**

-100.00 ≤ **x1, y1, x2, y2, r, delta** ≤ 100.00. Input akan memiliki 2 angka dibelakang titik.

Output

Baris pertama berisi keluaran dari fungsi DalamRange antara **p** dan **p1** dengan range **r**. 1 bila true, 0 bila false

Baris kedua berisi nilai point **p** setelah mengalami GlideReflection sesuai dengan arah **sbX** dengan pergeseran **delta**. Keluarkan dalam presisi 2 angka dibelakang titik dipisahkan oleh "," (**koma**) dan " " (**spasi**). Gunakan printf("...\n") untuk newline pada akhir output.

Contoh input output

Input	Output
5.00 6.00 11.00 12.00 17.00 1 2.00	1 7.00, -6.00
10.00 12.00 13.00 14.00 3.00 0 18.00	0 -10.00, 30.00

Kumpulkan ADT Point Anda dan drivernya(point2.h, point2.c, dan mpoint2.c) dengan penamaan sesuai standar praktikum. Zip ketiga file tersebut

Soal 3. (35%)

Implementasikan ADT GARIS berikut ini dengan memanfaatkan ADT POINT yang telah dihasilkan dari soal sebelumnya.

```
{ *** ADT LAIN YANG DIPAKAI *** }
USE POINT
{ *** Definisi TYPE GARIS *** }
type GARIS : < PAw : POINT, { Titik Awal }
PAkh : POINT { Titik Akhir } >
{Gunakan Epsilon = 0.00001 untuk mengecek presisi floating point}
{ ***** }
{ DEFINISI PRIMITIF }
{ ***** }
{ *** Konstruktor membentuk GARIS *** }
procedure MakeGARIS (input P1, P2 : POINT, output L : GARIS)
{ I.S. P1 dan P2 terdefinisi }
{ F.S. L terdefinisi dengan L.PAw= P1 dan L.Pakh=P2 }
{ Membentuk sebuah L dari komponen-komponennya }
{ Cara membaca: satu point satu baris, dengan atribut point dipisahkan oleh
spasi gunakan scanf("%f %f"... ) }
{ *** Selektor GARIS *** }
function GetPAw (G : GARIS) -> POINT
{ Mengirimkan komponen Titik Awal dari L GARIS }
function GetPAkh (G : GARIS) -> POINT
{ Mengirimkan komponen Titik Akhir dari L GARIS }
{ *** Set nilai komponen *** }
procedure SetPAw (input/output G : GARIS, inputnewPAw : POINT)
{ Mengubah nilai komponen PAw dari G }
procedure SetPAkh (input/output G : GARIS, inputnewPAkh : POINT)
{ Mengubah nilai komponen PAkh dari G }
{ ***** }
{ KELOMPOK INTERAKSI DENGAN I/O DEVICE, BACA/TULIS }
{ ***** }
procedure BacaGARIS (output L : GARIS)
{ I.S. sembarang }
{ F.S. L terdefinisi sebagai GARIS yang valid }
{ Proses : mengulangi membaca dua buah nilai P1 dan P2 sehingga dapat
membentuk GARIS yang valid. GARIS disebut valid jika titik awal tidak sama
dengan titik akhirnya. Bacalah 1 point per baris input, atribut point
dipisahkan oleh spasi, gunakan scanf("%f %f"... ) }
procedure TulisGARIS (input L : GARIS)
{ I.S. L terdefinisi }
{ F.S. L tertulis di layar dengan format ((x,y), (x,y)) beserta newline}
{ *** Kelompok operasi garis *** }
function PanjangGARIS (L : GARIS) -> real
{ Menghitung panjang garis L : berikan rumusnya }
function Gradien (L : GARIS) -> real
{ Mengembalikan Gradien (m) dari L }
{ Gradien garis yang melalui (x1,y1) dan (x2,y2) adalah: (y2-y1)/(x2-x1) }
procedure GeserGARIS (input/output L : GARIS, input deltaX, deltaY : real)
{ I.S. L terdefinisi }
{ F.S. L digeser sebesar deltaX dan ordinatnya sebesar deltaY }
{ Proses : PAw dan PAkh digeser! }
{ *** Kelompok predikat *** }
function IsTegakLurus (L1, L2 : GARIS) -> boolean
{ Menghasilkan true jika L1 tegak lurus terhadap L2 }
{ Dua garis saling tegak lurus jika hasil perkalian kedua gradiennya = -1 }
function IsSejajar (L1, L2 : GARIS) -> boolean
{ Menghasilkan true jika L "sejajar" terhadap L1 }
{ Dua garis saling sejajar jika memiliki gradien yang sama }
```

Garis.h dapat didownload di oddyseus. Dilarang merubah garis.h kecuali ada koreksi dari asisten.

Kumpulkan ADT Garis dan ADT Point dengan penamaan sesuai standar praktikum. (point.h point.c garis.h garis.c) dalam *.zip