

Problema Atena

Fișier de intrare `stdin`
Fișier de ieșire `stdout`

— Pamantul este platon.
— Pitagora a murit cu 75 înainte sa se nasca Pluton.

SC



Platon și Aristotel, buni prieteni, se rătăciseră în Atena. Se poate spune ca au o prietenie platonica. Totusi, se îndepărtasera asa mult încât nu se puteau auzi direct unul pe celălalt nici când strigau. Astfel, aveau nevoie de un intermediar: Pitagora. Când cei doi prieteni stau pe loc, Pitagora poate calcula folosind cunoștințele sale aria zonei în care trebuie să se afle ca să-i audă pe amândoi și le transmite informația prin codul Morse în formatul IEEE 754. Platon, bătrân fiind, nu se putea mișca la fel de repede ca discipolul său Aristotel, așa că a ales să rămână pe loc.

In aceasta problema tu te aflii in situatia lui Aristotel. Comisia are un cerc $\mathcal{C}_1$ ascuns de centru (X, Y) (care reprezinta pozitia lui Platon) și rază R (la care se aude ce spune filosoful). Concurantul trebuie să găsească numerele (X, Y, R) folosind cât mai puține întrebări de tipul:

Fie $\mathcal{C}_2$ un cerc de centru (x', y') și rază r' (representing Aristotle's position and his lung capacity). Care este aria intersecției cercurilor $\mathcal{C}_1$, respectiv $\mathcal{C}_2$. Unde x', y', r' sunt toate trei numere alese de concurent cu condițiile următoare:

- $-10^7 \leq x' \leq 10^7$
- $-10^7 \leq y' \leq 10^7$
- $0 < r' \leq 10^7$

Prin aria intersecției a două cercuri se înțelege aria porțiunii comune a discurilor determinate de aceste cercuri.

Cerință

Scrieți un program care ghicește folosind cât mai puține întrebări și cât mai precis coordonatele cercului. Punctajul se va acorda în funcție de numărul de întrebări și de precizia cu care este dat răspunsul.

Notă: Concurenților li se vor pune la dispoziție mai multe funcții geometrice ajutătoare cu scopul de a simplifica implementarea.

Protocol de interacțiune

Concurentul trebuie să implementeze exclusiv funcția

```
std::vector<double> solve();
```

Funcția va returna un vector $v = (v_1, v_2, v_3)$ de 3 elemente cu proprietatea că $X = v_1, Y = v_2, R = v_3$. Concurentul nu trebuie să implementeze funcția de `main()`

Pentru a adresa întrebări, concurentul poate apela funcția:

```
double query(double x, double y, double r);
```

Funcția va returna aria intersecției dintre cercul de centru (x, y) și rază r și cercul $\mathcal{C}_1$ ascuns de comisie.

Fișiere atașate

Participanții au acces la următoarele fișiere:

- `atena.h` – header-ul care definește funcțiile de interacțiune (`solve` și `query`) necesare pentru implementare.
- `atena.cpp` – un exemplu simplu de implementare care execută un `query` și răspunde cu cercul $(1, 2, 3)$ de fiecare dată.
- `lgrader.cpp` – un grader de testare pentru participanți care îi poate ajuta să verifice codul. Pentru punctarea unei submitii se va utiliza un alt grader.
- `geometry.h` – un fișier cu multiple funcții geometrie cu scop de a ușura implementarea concurenților.

Utilizarea lgrader în Code::Blocks

Pentru a utiliza `lgrader.cpp` în Code::Blocks:

1. Creați un proiect nou.
2. Adăugați un fișier denumit `solutie.cpp` (selectați `File > New > Empty File`), unde veți scrie codul soluției.
3. Adăugați un fișier gol numit `atena.h` și copiați conținutul din fișierul atașat `atena.h`
4. Copiați conținutul din `lgrader.cpp` în `main.cpp` pentru a simula rularea soluției.

După acest setup, scrieți în `solutie.cpp` implementarea funcției `solve`. Pentru a testa configurația, puteți copia inițial codul din `atena.cpp` în `solutie.cpp`. Apoi, apăsați **F9** sau selectați **Build and Run** pentru a compila și rula programul. La rulare, programul va citi valorile X, Y și R , și va simula interacțiunea dintre `lgrader` și soluția concurentului, afișând răspunsurile la interogările făcute prin `query` și validând răspunsul final oferit prin `solve`.

Date de ieșire

Se vor afișa 3 numere x, y, r reprezentând coordonatele cercului $\mathcal{C}_1$ ascuns.

Restricții

- Fie $\mathcal{C}_1$ cercul ascuns de comisie de centru (X, Y) și rază R .

- $-10^6 \leq X \leq 10^6$
- $-10^6 \leq Y \leq 10^6$
- **Atentie!** $10^4 \leq R \leq 10^6$

Punctare

Fie $D = (x - X)^2 + (y - Y)^2 + (r - R)^2$ marja de eroare a răspunsului și Q numărul de întrebări adresate de concurent.

Altfel, fie C_D și C_Q doi coeficienți de punctare definiți astfel:

$$C_D = \begin{cases} 1 & D < 10^{-5} \\ \frac{2 - \log_{10}(D)}{7} & 10^{-5} \leq D \leq 10^2 \\ 0 & D > 10^2 \end{cases}$$
$$C_Q = \begin{cases} 1 & Q \leq 20 \\ \left(\frac{70 - Q}{50}\right)^{1.3} \cdot 0.3 + 0.7 & 20 < Q \leq 70 \\ \frac{780 - 3Q}{1900} + 0.4 & 70 < Q \leq 260 \\ \frac{5000 - Q}{14960} + 0.15 & 260 < Q \leq 5000 \\ 0 & Q > 5000 \end{cases}$$

Atunci scorul vostru pentru un test va fi $100 \cdot C_D \cdot C_Q\%$ din maximul pe test.

Se garantează că pentru 20 de puncte centrul cercului comisiei se afla pe axa Ox.

Exemplu

Cercul ascuns este $\mathcal{C}_1 (X = 17.00000, Y = 2.62331, R = 6.7890)$.

Un exemplu de interacțiune este următorul:

```
query(1, 1.5, 2.5)
```

Aria intersecției dintre cercul ascuns $\mathcal{C}_1$ și cercul $(1, 1.5, 2.5)$ este 0.

```
query(15, 14, 7)
```

Aria intersecției dintre cercul $\mathcal{C}_1$ și cercul $(15, 14, 7)$ este 11.429000978.

```
return {17.000001, 2.62331, 6.7891};
```

Se observă că, deși valorile returnate nu coincid exact cu cele reale, soluția este considerată corectă deoarece marja de eroare este mai mică decât 10^{-5} .