مسألة أثينا

ملف المدخلات	stdin
ملف المخرجات	stdout

أفلاطون وأرسطو، أعز الأخوياء، ضيعوا في أثينا. تقدر تقول عنده صداقة أفلاطونية. لكنهم أبعدوا عن بعض حتى ما يقدرون يسمعون بعض لو صاحوا. هنا احتاجوا فزعة: فيثاغورس. إذا الاثنين كانوا واقفين، فيثاغورس بعلمه يقدر يحسب مساحة المكان اللي ينسمع صوتهم الاثنين ويرسل المعلومة بشفرة مورس بصيغة *IEEE* 754. أفلاطون، كان جد، ما يقدر يتحرك واجد زي خويه، قرر يبرك في مكانه.

أنت في مكان أرسطو في هذه المسألة. عند اللجنة دائرة مخفية $\mathscr{C}_1$ مركزها (X, Y) (تمثل موقع أفلاطون) وقطرها R (الحدود التي يسمع فيها صوت الفيلسوف). يجب على المتسابق إيجاد (X, Y, R) عن طريق سؤال أقل عدد من الأسئلة التالية:

عرف $\mathscr{C}_2$ بأنها دائرة مركزها (x', y') وقطرها r' (تمثل موقع أرسطو وسعة صدره). ما مساحة تقاطع $\mathscr{C}_1$ مع $\mathscr{C}_2$. حيث x', y', r' من اختيار المتسابق بشرط هذه الحدود:

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

مساحة تقاطع دائرتين هي المساحة المشتركة المغطاة بالقرصين معًا

المهمة

اكتب برنامج يخمن بأقل عدد أسئلة ممكنة وبأعلى دقة ممكنة يخمن الدائرة. الدرجة تحتسب بناءً على عدد الأسئلة ودقة الإجابة الناتجة.

ملاحظة: سيعطى المتسابقون عدة دوال هندسية لتبسيط كتابة البرنامج.

بروتوكول التواصل

يجب على المتسابق كتابة الدالة التالية:

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

الدالة ترجع مصفوفة (vector) $v = (v_1, v_2, v_3)$ من 3 عناصر حيث $X = v_1, Y = v_2, R = v_3$. المتسابق لا ينبغي أن يكتب الدالة الأساسية `main()`.

لسؤال الأسئلة، المتسابق يستدعي الدالة:

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

الدالة ترجع مساحة التقاطع بين دائرة مركزها (x, y) وقطرها r مع الدائرة المخفية $\mathscr{C}_1$ من اللجنة.

الملفات المرفقة

المتسابقون لديهم وصول للملفات التالية:

- `atena.h` – ملف يعرف أسماء دوال التواصل (`solve` و `query`) ضروري في الحل.
- `atena.cpp` – مثال بسيط لبرنامج يسأل سؤال واحد ويجاوب بالدائرة $(1, 2, 3)$ كل مرة.
- `lgrader.cpp` – مشغل لاختبار أكواد المتسابقين. برنامج مشغل آخر سيستعمل في التحكيم.
- `geometry.h` – ملف فيه العديد من الدوال الهندسية لتسهيل كتابة الكود.

طريقة استعمال lgrader في Code::Blocks

لتستعمل `lgrader.cpp` في Code::Blocks:

1. أنشئ مشروع جديد.
2. أضف ملف اسمه `solutie.cpp` (اختر `File > New > Empty File`)، حيث ستكتب الكود هناك.
3. أضف ملف فارغ اسمه `atena.h` وانسخ محتواه من الملف المرفق `atena.h`
4. انسخ محتوى `lgrader.cpp` للملف `main.cpp` لتشغل البرنامج

بعد هذا التجهيز، اكتب في `solutie.cpp` كود حل المسألة لاختبار التجهيز. تقدر مبدئيًا تنسخ كود `atena.cpp` للملف `solutie.cpp`. ثم تضغط F9 أو تحدد **Run and Build** لكي تبني وتشغل الكود. بعد التشغيل، البرنامج يقرأ القيم X و Y و R، ويحاكي التواصل بين lgrader وحل المتسابق، ويعرض الجوبة للأسئلة للاسئلة من query ويتأكد من الإجابة النهائية لدالة solve.

المخرجات

ثلاث أرقام x, y, r تمثل إحداثيات الدائرة المخفية $\mathscr{C}_1$.

الحدود

- عرف $\mathscr{C}_1$ بأنها الدائرة المخفية من اللجنة مركزها (X, Y) وقطرها R.
- $-10^6 \le X \le 10^6$

- $-10^6 \le Y \le 10^6$
- **ركز!** $10^4 \le R \le 10^6$

النقاط

عرف $D = (x - X)^2 + (y - Y)^2 + (r - R)^2$ بأنه هامش خطأ الإجابة، و Q عدد الأسئلة التي سألها المتسابق.

عرّف C_D و C_Q بأنها معاملين للدجة تعرّف كالتالي:

$$C_D = \begin{cases} 1 & D < 10^{-5} \\ \dfrac{2 - \log_{10}(D)}{7} & 10^{-5} \le D \le 10^2 \\ 0 & D > 10^2 \end{cases}$$

$$C_Q = \begin{cases} 1 & Q \le 20 \\ \dfrac{70 - Q}{250} + 0.8 & 20 < Q \le 70 \\ \dfrac{260 - Q}{76} + 0.4 & 70 < Q \le 260 \\ \dfrac{5000 - Q}{14960} + 0.15 & 260 < Q \le 500 \\ 0 & Q > 5000 \end{cases}$$

درجتك لاختبار واحد هي $100 \cdot C_D \cdot C_Q\%$ من أقصى درجة لهذا الاختبار.

مضمون أن 20 درجة من النقاط يكون مركز الدائرة المخفية على محور `Ox`.

مثال على التواصل

الدائرة المخفية هي $\mathscr{C}_1(X = 17.00000, Y = 2.62331, R = 6.7890)$. هذه الحالة غير ممكنة أن تكون مثال حيث R لا يوافق الحد الأدنى 10^4.

مثال للتواصل كالتالي:

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

مساحة التقاطع بين الدائرة المخفية $\mathscr{C}_1$ والدائرة $(1, 1.5, 2.5)$ هي 0.

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

مساحة التقاطع بين الدائرة المخفية $\mathscr{C}_1$ والدائرة $(15, 14, 7)$ هي 11.429000978.

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

يلاحظ أن بالرغم أن القيم المرجعة لا تتطابق مع القيم الحقيقية، الحل يعتبر صحيح لأن هامش الخطأ أقل من 10^{-5}.