

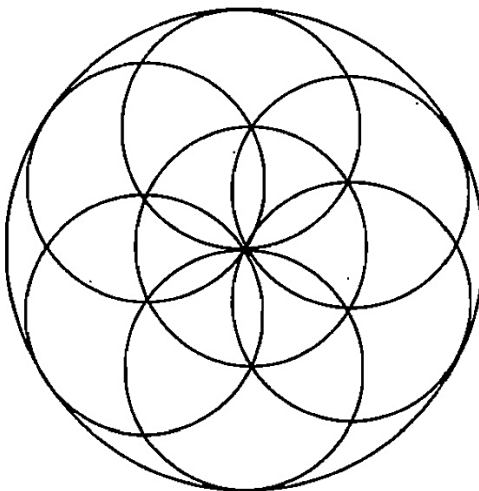
ΚΥΠΡΙΑΚΟΣ ΣΥΝΔΕΣΜΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
CYPRUS COMPUTER SOCIETY



ΠΑΓΚΥΠΡΙΑ ΟΛΥΜΠΙΑΔΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
CYPRUS OLYMPIAD IN INFORMATICS

Gandalf

Στη Μέση Γη, όπου οι αλγόριθμοι είναι μαγεία και τα δεδομένα μυστικά, ο Γκάνταλφ ο Γκρίζος αντιμετωπίζει μια νέα δοκιμασία. Ο σκοτεινός άρχοντας Σάουρον έχει κρύψει τους Κύκλους της Δύναμης σε άγνωστες τοποθεσίες ψηλά στον ουρανό, και με αυτούς ελέγχει την ισορροπία του κόσμου.



Εικόνα 1: Πόσους κύκλους βλέπετε;

Για να τους ανακαλύψει, ο Γκάνταλφ ο Γκρίζος έχει στην διάθεσή του το Ραβδί των Αρχαίων. Όταν το στρέφει προς τον ουρανό μέσα στην έναστρη νύχτα, του αποκαλύπτει τις συντεταγμένες (x, y) ενός τυχαίου αστεριού που βρίσκεται πάνω στην περιφέρεια ενός τυχαίου κύκλου. Δυστυχώς όμως το ραβδί έχει μόνο 100,000 χρήσεις πριν εξασθενήσει η μαγεία του.

Η δική σας αποστολή είναι να βοηθήσετε τον Γκάνταλφ, ανακαλύπτοντας τα κέντρα και τις ακτίνες όλων (K) των Κύκλων της Δύναμης, χρησιμοποιώντας το ραβδί όσοες λιγότερες φορές γίνεται.

Διαδραστικότητα

Το πρόβλημα είναι διαδραστικό (interactive) και η λύση σας πρέπει να υλοποιήσει τη συνάρτηση:

```
void solve(int k);
```

Η υλοποίηση μπορεί να χρησιμοποιήσει τις παρακάτω συναρτήσεις από το παρεχόμενο αρχείο `circle.h`:

- `pair<double, double> sample_star();`

Καλείτε τη συνάρτηση `sample_star` για να λάβετε τις συντεταγμένες ενός αστεριού (x, y) , το οποίο είναι ένα τυχαίο σημείο πάνω σε έναν από τους K κύκλους. Αυτό αντιπροσωπεύει μια χρήση του μαγικού ραβδιού.

- `void answer(double cen_x, double cen_y, double radius);`

Καλείτε τη συνάρτηση `answer` για να αναφέρετε τις παραμέτρους ενός κύκλου που βρέθηκε. Αυτή η συνάρτηση πρέπει να κληθεί ακριβώς K φορές.

Περιορισμοί

- Το πλήθος των κύκλων: $1 \leq K \leq 20$.
- Μπορείτε να καλέσετε τη `sample_star` μέχρι **100,000** φορές.
- Τα κέντρα (x_i, y_i) και οι ακτίνες (r_i) όλων των κύκλων είναι αριθμοί των οποίων η απόλυτη τιμή είναι μικρότερη ή ίση του **500,000**. Όλες οι παράμετροι που υποβάλλετε όταν καλείτε τη `answer` πρέπει επίσης να τηρούν αυτόν τον περιορισμό.
- Θεωρείστε δεδομένο ότι η ευκλείδεια απόσταση μεταξύ των κέντρων οποιωνδήποτε δύο κύκλων θα είναι μεγαλύτερη από **100**.

Σημείωση

Για δύο σημεία στο επίπεδο, $P = (p_x, p_y)$ και $Q = (q_x, q_y)$, η ευκλείδεια απόσταση είναι:

$$\sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}.$$

Υποπροβλήματα

Υποπρόβλημα	Πόντοι	Περιορισμός για K	Πρόσθετοι περιορισμοί
1	3	$K = 1$	Κανένας
2	7	$K = 2$	Οι κύκλοι δεν τέμνονται
3	20	$K = 2$	Κανένας
4	13	$K = 5$	Κανένας
5	20	$K = 10$	Κανένας
6	37	$K \leq 20$	Κανένας

Μέρος των πόντων για κάθε υποπρόβλημα βασίζεται στο ελάχιστο αποτέλεσμα των αρχείων ελέγχου του υποπροβλήματος.

Λεπτομέρειες αξιολόγησης

Το αποτέλεσμα για κάθε τεστ υπολογίζεται ως εξής:

1. Αν υπάρχει κύκλος για τον οποίο η απόλυτη διαφορά μεταξύ της πρότασής σας και των πραγματικών παραμέτρων υπερβαίνει το 0.001, θα λάβετε 0 πόντους.
2. Διαφορετικά, το αποτέλεσμα υπολογίζεται με τον τύπο:

$$0.3 + 0.7 \times \sqrt{\frac{500}{\max(Q, 500)}}$$

όπου Q είναι ο αριθμός των κλήσεων στη `sample_star`.

Βαθμολογητής (Grader)

Παρέχεται ένα παράδειγμα βαθμολογητή (`Lgrader.cpp`), όπου κάθε αστέρι δημιουργείται ως εξής:

1. Επιλέγεται ένας τυχαίος κύκλος από τους K κύκλους.
2. Έστω το κέντρο του είναι (c_x, c_y) και η ακτίνα r .
3. Επιλέγεται μια τυχαία γωνία θ στο εύρος $[\theta, 2\pi)$.
4. Υπολογίζονται οι συντεταγμένες του σημείου:

$$x = c_x + r * \cos(\theta)$$

$$y = c_y + r * \sin(\theta)$$

5. Επιστρέφεται το σημείο (x, y).

Αυτή η διαδικασία δημιουργεί ένα τυχαίο σημείο στην περιφέρεια ενός τυχαίου κύκλου. Επειδή οι υπολογισμοί γίνονται με αριθμούς κινητής υποδιαστολής, εγγυόμαστε ότι για κάθε σημείο που δημιουργείται με τη `sample_star`, η μικρότερη απόσταση από την περιφέρεια είναι $< 10^{-4}$.

Παράδειγμα διαδραστικότητας

Έστω $K = 2$ και οι κύκλοι είναι:

- Κύκλος 1: κέντρο (100, 200), ακτίνα 50
- Κύκλος 2: κέντρο (-150, -300), ακτίνα 75

Παράδειγμα διαδραστικότητας:

Ο βαθμολογητής καλεί: `solve(2);`

Εσείς καλείτε τη `sample_star` στην υλοποίηση της `solve`:

```
sample_star() -> (50.4373..., 206.5988...)
sample_star() -> (-79.3886..., -274.7206...)
```

Εσείς καλείτε τη `answer` στην υλοποίηση της `solve`:

```
answer(100.0, 200.0, 50.0);
answer(-150.0, -300.0, 75.0);
```

Στο τέλος της εκτέλεσης, ο βαθμολογητής ελέγχει τις απαντήσεις.

Μπορείτε να οπτικοποιήσετε τα αρχεία ελέγχου χρησιμοποιώντας το πιο κάτω εργαλείο:

<https://gandalf.cmscoinformatics.org/>