

Problemy dżdżownicy

Szukasz miejsca w ziemi aby umieścić tam swoją dżdżownicę, Maximusa. Poszukiwania ograniczyłeś do obszaru w kształcie pudełka (prostopadłościanu) o wymiarach $N \times M \times K$ centymetrów. Pudełko podzieliłeś na trójwymiarową kratę jednocentymetrowych pól, oznaczonych pozycją w tej kratce (x, y, z) ($1 \leq x \leq N$, $1 \leq y \leq M$, $1 \leq z \leq K$). Każde pole ma określoną wilgotność $H[x, y, z]$, która jest liczbą całkowitą z zakresu $1 \dots 10^9$. Możesz zmierzyć wilgotność w danym polu używając specjalistycznego urządzenia.

Maximus preferuje wilgotne miejsca, zatem musisz umieścić go w polu, które jest co najmniej tak wilgotne jak wszystkie sąsiadujące pola, w przeciwnym wypadku Maximus ucieknie i będziesz musiał go szukać. Innymi słowy, musisz umieścić Maximusa w lokalnym maksimum. Konkretniej, musisz znaleźć pole (x, y, z) , takie że

$$H[x, y, z] \geq \max(H[x + 1, y, z], H[x - 1, y, z], H[x, y + 1, z], H[x, y - 1, z], H[x, y, z + 1], H[x, y, z - 1]),$$

gdzie wartość pola poza pudełkiem to 0 (ponieważ Maximus absolutnie chce zostać w tym pudełku).

Jednakże, liczba pól może być bardzo duża, zatem nie chcesz mierzyć wilgotności każdego z nich. Z tego powodu, w tym zadaniu będziesz komunikował się z programem sprawdzającym, pytając o wilgotność pewnych punktów. Kiedy już znajdziesz odpowiednie miejsce dla Maximusa, podaj tę lokalizację programowi sprawdzającemu.

Interakcja

W pierwszym wierszu standardowego wejścia znajdują się cztery dodatnie liczby całkowite: N , M , K i Q , gdzie N , M oraz K to rozmiary obszaru poszukiwań, a Q jest maksymalną liczbą pomiarów, które możesz wykonać.

Następnie możesz wypisać co najwyżej Q wierszy postaci $? x y z$ na standardowe wyjście. Taki wiersz oznacza zapytanie o wilgotność w polu (x, y, z) . Dla każdego takiego wiersza program sprawdzający w odpowiedzi wypisze pojedynczy wiersz z liczbą całkowitą $H[x, y, z]$, którą Twój program może wczytać ze standardowego wejścia.

Po zadaniu wszystkich zapytań Twój program musi wypisać dokładnie jeden wiersz postaci $! x y z$ i się zakończyć. Taki wiersz oznacza, że pole (x, y, z) jest odpowiednią lokalizacją dla Maximusa według powyższych kryteriów. Program sprawdzający nie zwróci żadnej odpowiedzi dla takiego komunikatu.

Wszystkie wartości x, y, z muszą spełniać $1 \leq x \leq N$, $1 \leq y \leq M$, $1 \leq z \leq K$. Jeżeli tak nie będzie, albo pewien wiersz będzie miał niewłaściwy format, bądź zadasz więcej niż Q zapytań, program sprawdzający wypisze -1 oraz zakończy interakcję. W takim przypadku Twój program też powinien się zakończyć. W przeciwnym wypadku możesz niepoprawnie otrzymać werdykt Runtime Error albo Time Limit Exceeded.

Pamiętaj, żeby opróżnić bufor wyjściowy *po każdym wierszu*, przed wczytaniem odpowiedzi sprawdzarki, inaczej twój program otrzyma wynik Time Limit Exceeded. Komendy opróżniające bufor we wspieranych językach:

- Java: `System.out.println()` robi to automatycznie.
- Python: `print()` również robi to sam.
- C++: `cout << endl`; opróżnia bufor, dodatkowo wypisując znak nowej linii. Jeżeli używasz `printf: fflush(stdout)`.
- Pascal: `Flush(Output)`.

Aby pomóc z radzeniem sobie z komunikacją, zapewniamy opcjonalny dodatkowy kod, który możesz skopiować do swojego programu. Odnośnik do tego kodu dla wszystkich wspieranych języków (C++, Pascal, Java, Python) możesz znaleźć w menu bocznym na stronie z problemami. Pomocniczy kod używa zoptymalizowanych metod wczytywania i wypisywania, co może być przydatne dla Javy i Pythona w dwóch ostatnich przypadkach testowych.

Program sprawdzający *nie będzie adaptacyjny*, tj. dla każdego testu będą ustalone wartości wilgotności, które nie będą zależeć od pomiarów wykonanych przez Twój program.

Ograniczenia

Zestaw testów dzieli się na kilka grup, każda jest warta pewną liczbę punktów. Każda grupa składa się z jednego bądź większej liczby testów. Aby otrzymać punkty za daną grupę, Twoje rozwiązanie musi przejść wszystkie testy z tej grupy. Ostateczny wynik za zadanie jest liczony jako maksymalny wynik z pojedynczych zgłoszeń.

Grupa	Punkty	Limity
1	10	$M = K = 1, N = 1\,000\,000, Q = 10\,000$
2	22	$M = K = 1, N = 1\,000\,000, Q = 35$
3	12	$K = 1, N = M = 200, Q = 4\,000$
4	19	$K = 1, N = M = 1\,000, Q = 3\,500$
5	14	$N = M = K = 100, Q = 100\,000$
6	23	$N = M = K = 500, Q = 150\,000$

Sample dialogue

Na Kattisie znajduje się jeden test przykładowy. W tym teście pudełko ma wymiary $3 \times 1 \times 1$, wilgotność kolejnych pól wynosi odpowiednio $\{10, 14, 13\}$. Poniżej znajduje się przykład interakcji dla tego testu. Linie oznaczone YOU zostały wypisane przez Twój program, a linie oznaczone JUDGE przez Kattis (wczytane przez Twój program).

14 jest większe bądź równe sąsiednim wartościom (10 i 13), pozycja (2, 1, 1) jest doskonałym miejscem dla Maximusa. Twój program użył łącznie trzech zapytań, co było maksymalną możliwą liczbą zapytań w tym teście. Program otrzyma wynik Accepted.

```
JUDGE: 3 1 1 3
YOU:    ? 3 1 1
JUDGE: 13
YOU:    ? 2 1 1
JUDGE: 14
YOU:    ? 1 1 1
JUDGE: 10
YOU:    ! 2 1 1
```

Przykładowe wejście 1

```
3 1 1 3 123123 fixed 10 14 13
```

Przykładowe wyjście 1

```
blah
```