

Неопределено

Тази година N състезатели участват в Stelele Informaticii, номерирани с числата от $0, 1, \dots, N - 1$.

След състезанието състезателите са класирани спрямо скритата пермутация P на числата $0, 1, \dots, N - 1$, където $P[i]$ е позицията на i -тия състезател в крайното класиране. Позиция 0 съответства на първото място. Алек Сдима знае крайното класиране, а Вие целите да го узнаете.

Може да предположите класиране Q , което трябва също да е пермутация на числата $0, 1, \dots, N - 1$, където $Q[i]$ е позицията, която давате на i -тия състезател. Грешката при състезател i във Вашето класиране дефинираме като $Q[i] - P[i]$.

След като предложите класиране, Алек ще ви каже максималната грешка за някой състезател, или по-формално

$$\max_{0 \leq i < N} (Q[i] - P[i]).$$

Вашата задача е да определите крайно класиране P , правейки колкото се може по-малко предположения за класирането.

Имплементация

Трябва да имплементирате следната функция.

```
std::vector<int> find_ranking(int N)
```

- N : броят на състезателите.
- Тази функция трябва да върне скритата пермутация P .
- Тази функция се извиква точно веднъж в теста.

Може да извиквате следната функция:

```
int ask(std::vector<int> Q)
```

- Q е пермутация на числата $0, 1, \dots, N - 1$.
- $Q[i]$ представлява позицията на i -тия състезател.

- Тази функция връща

$$\max_{0 \leq i < N} (Q[i] - P[i]).$$

- Функцията може да бъде извикана най-много 20 000 пъти в един тест.

Грейдърът е **адаптивен**. Това означава, че грейдърът няма фиксирана пермутация P . Вместо това, стойностите, върнати от функцията `ask` могат да зависят от миналите извиквания на `ask`.

Гарантирано е, че грейдърът отговаря по такъв начин, че след всяко извикване на функцията има поне една пермутация P , консистентна с всички отговори на заявките досега.

Ограничения

- $1 \leq N \leq 1000$
- P е пермутация на числата $0, 1, \dots, N - 1$.

Подзадачи и оценяване

Ако някое от извикванията към функцията `ask` не отговаря на описаните правила по-горе или пермутацията, върната от `find_ranking` е грешна, оценката ви за теста ще бъде 0.

В противен случай, нека Q да бъде максималният брой извиквания на `ask` в теста. Допълнително, нека X да е равно на

$$X = \min \left(1, \frac{2}{1 + \left(\frac{Q}{1997} \right)^{3/2}} \right).$$

Тогава резултатът Ви е изчислен по следния начин:

Подзадача	Точки	Допълнителни ограничения
1	10	$N \leq 100$
2	$90 \cdot X$	$N = 1000$

В частност, за да получите 100 точки, трябва да използвате най-много 1 997 извиквания на `ask` във всеки тест. Оценката за всяка подзадача се закръгля до две цифри след десетичната запетая.

Пример

Нека $N = 4$ и тайното класиране да е $P = [2, 0, 3, 1]$. Това означава, че състезател 1 е на първо място, състезател 3 е на второ място, състезател 0 е на трето място и състезател 2 е на четвърто място.

Нека разгледаме следното извикване на функцията:

```
ask({0, 1, 2, 3})
```

За това класиране стойностите на $Q[i] - P[i]$ са $[-2, 1, -1, 2]$. Максималната грешка е 2 и функцията връща точно 2.

Нека сега разгледаме:

```
ask({3, 0, 2, 1})
```

Стойностите $Q[i] - P[i]$ са $[1, 0, -1, 0]$, така че функцията връща 1.

Накрая, ако предложеното класиране е същото като крайното,

```
ask({2, 0, 3, 1})
```

всички стойности на $Q[i] - P[i]$ са 0 и функцията връща 0.

Примерен грейдър

Примерният грейдър **не е адаптивен**. Тайната пермутация P е фиксирана преди `find_ranking` да е извикана.

Примерният грейдър приема входа по следния формат:

- ред 1: N
- ред 2: $P[0] P[1] \dots P[N - 1]$

За всяко извикване към `ask(Q)`, примерният грейдър връща

$$\max_{0 \leq i < N} (Q[i] - P[i]).$$

Примерният грейдър връща изход в следния формат:

- ред 1: C
- ред 2: $R[0] R[1] \dots R[N - 1]$

Тук C е броят извиквания на `ask` и R е пермутацията, върната от `find_ranking`.