



# Rzutki

Limit pamięci: 512 MB

Jak powszechnie wiadomo, w Bajtocji rzutki są sportem narodowym, a niemal w każdym domu wisi klasyczna tarcza złożona z  $n$  celów, ponumerowanych kolejno od 1 do  $n$ . Dla dwóch celów o numerach  $i$  oraz  $j$  definiujemy odległość jako

$$d(i, j) = |i - j|.$$

Bajtek i Bitek postanowili rozegrać nietypową partię. Będą wykonywać kolejne rzuty tak długo, aż wszystkie cele zostaną trafione (czyli łącznie zostanie rzuconych dokładnie  $n$  rzutek). Obowiązują przy tym następujące dwie zasady:

1. Każda kolejna rzutka musi trafić w cel, w który żadna wcześniejsza rzutka jeszcze nie trafiła.
2. Każda oprócz pierwszej rzutki musi trafić w cel położony jak najdalej od już trafionych celów. Formalnie: niech  $S$  oznacza zbiór celów już trafionych, a  $x$  będzie rozważanym (jeszcze nie trafionym) celem. Definiujemy wartość

$$\text{odl}(x, S) = \min_{y \in S} d(x, y),$$

czyli odległość od  $x$  do najbliższego już trafionego celu (dla pierwszego rzutu, gdy  $S$  jest puste, warunek wyboru celu nie narzuca ograniczeń). Gracz musi wybrać taki cel  $x$ , aby wartość  $\text{odl}(x, S)$  była **maksymalna** spośród wszystkich jeszcze nie trafionych celów. Jeżeli istnieje wiele celów spełniających powyższy warunek, gracz może wybrać **dowolny** z nich.

Bajtek ma swoją ulubioną rzutkę: jest to  $a$ -ta rzutka rzucona w całej grze (licząc wszystkie rzuty od 1 do  $n$ ). Ma też ulubiony cel o numerze  $b$ . Bajtek zastanawia się, czy da się tak dobierać cele, aby  $a$ -ta rzutka trafiła w cel  $b$ .

## Wejście

W pierwszym i jedynym wierszu wejścia znajdują się trzy liczby całkowite  $n$ ,  $a$  i  $b$  ( $1 \leq n \leq 500\,000$ ,  $1 \leq a, b \leq n$ ), oznaczające odpowiednio: liczbę celów na tarczy, numer ulubionej rzutki Bajtka oraz numer jego ulubionego celu.

## Wyjście

Jeśli nie istnieje żadna kolejność rzutów spełniająca warunki zadania, w jedynym wierszu wyjścia należy wypisać  $-1$ .

W przeciwnym wypadku należy wypisać ciąg  $n$  liczb całkowitych  $x_1, x_2, \dots, x_n$  ( $1 \leq x_i \leq n$ ), gdzie  $x_i$  oznacza numer celu trafionego przez  $i$ -tą rzutkę. Wypisany ciąg musi opisywać poprawny przebieg gry i spełniać warunek  $x_a = b$ .

Jeżeli istnieje wiele poprawnych rozwiązań, należy wypisać **najmniejsze leksykograficznie**.

# Rzutki

Limit pamięci: 512 MB

## Przykłady

Wejście dla testu r2e0a:

```
1 1 1
```

Wyjście dla testu r2e0a:

```
1
```

**Wyjaśnienie:** Na tarczy jest tylko jeden cel. Pierwsza (i jedyna) rzutka musi trafić w cel 1, więc warunek  $x_1 = b$  jest spełniony.

Wejście dla testu r2e0b:

```
3 2 2
```

Wyjście dla testu r2e0b:

```
-1
```

**Wyjaśnienie:** Chcemy, aby druga rzutka trafiła w cel 2. Rozważmy wszystkie możliwości pierwszego rzutu:

- Jeśli pierwsza rzutka trafi w 2, to druga nie może już trafić w 2 (cel musi być nowy).
- Jeśli pierwsza rzutka trafi w 1, to dla pozostałych celów mamy:

$$\text{odl}(2, \{1\}) = 1, \quad \text{odl}(3, \{1\}) = 2,$$

więc druga rzutka musi trafić w 3 (bo jest najdalej).

- Jeśli pierwsza rzutka trafi w 3, analogicznie druga rzutka musi trafić w 1.

W żadnym przypadku druga rzutka nie może trafić w cel 2, więc rozwiązanie nie istnieje.

Wejście dla testu r2e0c:

```
5 3 2
```

Wyjście dla testu r2e0c:

```
4 1 2 3 5
```

## Pozostałe testy przykładowe

- test r2e0d:  $n = 20, a = 7, b = 13$ .
- test r2e0e:  $n = 100\,000, a = 50\,000, b = 1$ .
- test r2e0f:  $n = 500\,000, a = 250\,000, b = 400\,000$ .

## Ocenianie

Limit pamięci dla Pythona wynosi 1024 MB.

Podzadanie	Ograniczenia	Limit czasu	Punkty
1	$a = 1, n \leq 1\,000$	1 s (C++) / 4 s (Python)	4
2	$a = 1$	4 s (C++) / 120 s (Python)	10
3	$a = 2$	6 s (C++) / 120 s (Python)	5
4	$n \leq 20$	1 s (C++) / 8 s (Python)	10
5	$n \leq 100$	1 s (C++) / 8 s (Python)	17
6	$n \leq 1\,000$	1 s (C++) / 12 s (Python)	14
7	$n \leq 100\,000$	6 s (C++) / 120 s (Python)	25
8	Brak dodatkowych ograniczeń	7 s (C++) / 120 s (Python)	15