

# Sekwencjonowanie BNA (A)

Limit pamięci: 256 MB

Limit czasu: 10.00 s

To zadanie jest interaktywne. Po wypisaniu każdego wiersza należy opróżnić bufor. W C++ możesz użyć `cout « flush`, w Java – `System.out.flush()`, w Python – `sys.stdout.flush()`. Pamiętaj o wypisaniu białego znaku podczas opróżniania bufora. Należy ściśle trzymać się protokołu opisanego w rozdziale *Protokół interakcji* – niewczytanie wiadomości wysłanej przez interaktor skutkuje werdyktem *wrong answer*.

Rozpoczęte zostały prace nad opracowaniem leku na bitozę – chorobę genetyczną prześladowaną mieszkańców Bitocji od paru dobrych wieków. Ostatnio dokonano pewnego postępu w badaniach. Bitoccy naukowcy podejrzewają, że udało im się wyizolować segment BNA odpowiedzialny za wywoływanie choroby.

BNA (kwas bitrybonukleinowy) jest organicznym związkiem chemicznym, który koduje informację genetyczną. BNA może być opisane jako ciąg kodujących go kwasów bitowych: zerozyny ("0") oraz jedyniny ("1"). Aktualnym problemem w kontynuowaniu badań jest poznanie ciągu kwasów bitowych segmentu BNA, który odpowiada za bitozę.

Bajtazar – jako osoba odpowiedzialna za nadzór nad badaniami – poprosił ciebie, znajomego bioinformatyka, o napisanie programu komputerowego, który odczyta sekwencję BNA kodującego bitozę. Aby ci to umożliwić, Bajtazar udostępnił ci technologię pozwalającą na przeprowadzenie testów stwierdzających, czy sekwencja syntetycznie wytworzonego BNA znajduje się jako **spójny podciąg** sekwencji BNA bitozy. Niestety, przeprowadzenie pojedynczego testu może trwać nawet parę godzin, dlatego zostałeś poproszony o to, aby twój program wykonał jak najmniej testów.

## Interakcja

Na początku w pierwszym wierszu wejścia znajdzie się liczba całkowita  $N$ , oznaczająca długość sekwencji BNA bitozy.

Następnie należy przeprowadzić pewną (być może zerową) liczbę testów przy pomocy zapytań. Zapytania mają postać  $? A$ , gdzie  $A$  jest ciągiem znaków 0 oraz 1 oznaczającym sekwencję syntetycznego BNA do zastosowania w teście. Odpowiedzią na zapytanie jest 1, jeśli sekwencja syntetycznego BNA znajduje się jako spójny podciąg w sekwencji BNA odpowiedzialnej za bitozę; w przeciwnym wypadku interaktor wypisze 0.

Po wykonaniu wszystkich zapytań, należy wypisać sekwencję BNA bitozy  $B$  w formacie:  $! B$ .

Wypisanie sekwencji BNA bitozy nie wlicza się do liczby zapytań.

**Interaktor jest adaptacyjny**, oznacza to, że sekwencja BNA bitozy **nie musi być ustalona z góry** i może być dobierana na bieżąco w trakcie działania programu, o ile jej zmiana nie zaprzeczy wynikom wykonanych do tej pory zapytań.

## Punktacja

We wszystkich testach zachodzi:  $1 \leq N \leq 10\,000$ .

Liczba punktów **zależy od liczby zadanych zapytań**.

Oznaczmy przez  $Q$  maksymalną liczbę zapytań zadanych przez program. Wówczas, liczba punktów zależy w następujący sposób od  $Q$ :

Próg $Q$	Liczba punktów
30 000	10
20 000	20
15 000	40
11 000	60
10 200	70
10 035	85
10 025	90
10 015	100

Jeśli  $Q \geq 30\,000$ , to otrzymasz 0 punktów za to zadanie.

W przeciwnym wypadku, gdy  $Q \leq 10\,015$ , otrzymasz 100 punktów.

W pozostałych przypadkach, jeśli  $Q$  jest równe któremuś z powyższych progów w tabeli, otrzymasz przypisaną mu liczbę punktów.

W przeciwnym razie  $Q$  znajduje się pomiędzy dwoma progami i otrzymasz liczbę punktów wyznaczoną liniową interpolacją pomiędzy nimi, z zaokrągleniem w dół.

## Przykładowa interakcja

Wejście	Wyjście
---------	---------

4	
---	--

	? 0
--	-----

1	
---	--

	? 00
--	------

0	
---	--

	? 1
--	-----

1	
---	--

	? 11
--	------

0	
---	--

	? 0101
--	--------

0	
---	--

	? 00000
--	---------

0	
---	--

	! 1010
--	--------

Powyższa interakcja została przeprowadzona poprawnie. W szczególności zapytanie o długości większej niż  $N$  jest dopuszczalne. W powyższej interakcji  $Q$  jest równe 6, a więc  $Q \leq 10\,015$ , co oznacza, że za powyższy test zostanie przydzielone 100 punktów.

## Narzędzia do testowania lokalnego

Zostaną udostępnione testy przykładowe i program **interaktor.cpp** oraz przykładowe błędne rozwiązanie o poprawnym schemacie komunikacji **wa.cpp**. Żeby uruchomić rozwiązanie na teście, należy skompilować rozwiązanie i interaktorkę, a następnie użyć komendy:

```
[plik wykonywalny z interaktor.cpp] [test] [plik wykonywalny z rozwiązaniem]
```

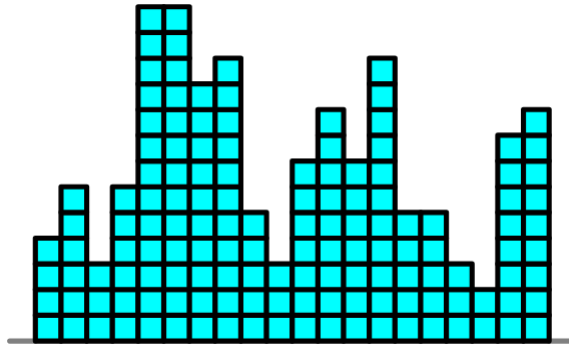
```
Na przykład: ./interaktor 0 ./wa
```

# Klocki (B)

Limit pamięci: 256 MB

Limit czasu: 2.00 s

Bajtazar postanowił przygotować na wystawę artystyczną instalację złożoną z podświetlanych klocków. Ustawił je w  $N$  stosów, ponumerowanych od 1 do  $N$ . Wysokość  $i$ -tego stosu oznaczmy przez  $S_i$ .



Przykładowe ustawienie klocków dla  $N = 20$

Komisja bezpieczeństwa festiwalu stwierdziła jednak, że instalacja może być niebezpieczna. Uznano, że dzieło jest bezpieczne tylko wtedy, gdy wysokości sąsiednich stosów nie różnią się o więcej niż  $H$  klocków, tzn. dla każdego  $1 \leq i < N$  zachodzi

$$|S_i - S_{i+1}| \leq H.$$

Bajtazar nie chce usuwać swojej instalacji, ale może ją zmodyfikować. W jednej operacji może:

- dodać jeden klocek na szczyt wybranego stosu,
- usunąć jeden klocek z wierzchu wybranego stosu (o ile ten stos nie jest pusty).

Bajtazar chce wykonać jak najmniej operacji, aby jego instalacja stała się bezpieczna. Twoim zadaniem jest obliczenie minimalnej liczby potrzebnych operacji.

## Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite  $N$  oraz  $H$  – odpowiednio liczba stosów oraz maksymalna dozwolona różnica wysokości sąsiednich stosów.

W drugim (i ostatnim) wierszu wejścia znajduje się  $N$  nieujemnych liczb całkowitych oddzielonych pojedynczymi spacjami – wysokości  $S_1, S_2, \dots, S_N$  kolejnych stosów.

## Wyjście

W pierwszym i jedynym wierszu wyjścia powinna znaleźć się pojedyncza liczba całkowita – minimalna liczba operacji, które musi wykonać Bajtazar, aby jego instalacja stała się bezpieczna.

## Ograniczenia

$1 \leq N \leq 200\,000$ ,  $0 \leq H \leq 10^9$ ,  $0 \leq S_i \leq 10^9$  dla każdego  $1 \leq i \leq N$

## Podzadania

Podzadanie	Warunki	Punkty
1	$N \leq 10$ , $S_i \leq 4$	3
2	$N \leq 14$ , $H \leq 1$ , $S_i \leq 4$	4
3	$N \leq 10$ , $H \leq 2$	9
4	$H = 0$	5
5	$N \leq 500$ , $S_i \leq 400$	6

Podzadanie	Warunki	Punkty
6	$N \leq 500, S_i \leq 5000$	11
7	$N \leq 5000, S_i \leq 5000$	11
8	$N \leq 5000$	22
9	brak dodatkowych ograniczeń	29

## Przykład

### Wejście

6 1  
2 10 0 2 4 3

### Wyjście

10

### Wyjaśnienie

Bajtazar może zabrać 7 klocków z drugiego stosu, następnie dodać 2 klocki na trzeci stos, oraz 1 klocek na czwarty stos.

### Wejście

6 3  
2 10 2 6 4 3

### Wyjście

6

### Wejście

4 1  
1 4 1 4

### Wyjście

4

### Wejście

10 1  
10 9 8 7 6 5 4 3 2 1

### Wyjście

0

### Wejście

3 0  
1 1 3

### Wyjście

2

# Urodzinowy XOR (c)

Limit pamięci: 256 MB

Limit czasu: 2.50 s

Jasiu dostał na urodziny ciąg  $N$  liczb  $A_1, A_2, \dots, A_n$  i zastanawia się teraz, jak bardzo powinien się cieszyć z tego prezentu.

Postanowił, że swoją decyzję podejmie na podstawie wartości  $XOR$  wszystkich liczb postaci  $A_i + A_j$  dla  $1 \leq i \leq j \leq N$ .

Jak zazwyczaj, Jasiu nie poradził sobie samemu z tym zadaniem i poprosił Cię o pomoc w znalezieniu interesującej go wartości.

## Wejście

W pierwszym wierszu wejścia znajduje się pojedyncza liczba całkowita  $N$  – długość ciągu Jasia.

W drugim (i ostatnim) wierszu wejścia znajduje się  $N$  liczb całkowitych oddzielonych pojedynczymi spacjami – ciąg  $A_1, A_2, \dots, A_n$  Jasia.

## Wyjście

W pierwszym i jedynym wierszu wyjścia powinna znaleźć się pojedyncza liczba całkowita – wartość interesująca Jasia.

## Ograniczenia

$1 \leq N \leq 10^6$ ,  $1 \leq A_i \leq 5 \cdot 10^8$ .

## Podzadania

Podzadanie	Warunki	Punkty
1	$N \leq 4\,000$	7
2	$A_i \leq 4\,000$	11
3	$A_i \leq 10^6$	21
4	$N \leq 10^5$	38
5	brak dodatkowych ograniczeń	23

## Przykład

### Wejście

4  
3 7 10 7

### Wyjście

17

### Wyjaśnienie

$A_1 + A_1 = 3 + 3 = 6$   
 $A_1 + A_2 = 3 + 7 = 10$   
 $A_1 + A_3 = 3 + 10 = 13$   
 $A_1 + A_4 = 3 + 7 = 10$   
 $A_2 + A_2 = 7 + 7 = 14$   
 $A_2 + A_3 = 7 + 10 = 17$   
 $A_2 + A_4 = 7 + 7 = 14$   
 $A_3 + A_3 = 10 + 10 = 20$   
 $A_3 + A_4 = 10 + 7 = 17$   
 $A_4 + A_4 = 7 + 7 = 14$   
 $6 \oplus 10 \oplus 13 \oplus 10 \oplus 14 \oplus 17 \oplus 14 \oplus 20 \oplus 17 \oplus 14 = 17$

### Wejście

4  
9 9 9 9

### Wyjście

0

**Wejście**

10

21 7 4 19 30 1 15 12 28 27

**Wyjście**

56