



Park rozrywki

Limit pamięci: 256 MB

Bajtek pojechał razem ze swoją rodziną na wakacje za granicę, do Bitocji. Główną atrakcją wyjazdu miał być Bitocki park rozrywki. Przy czym słowa "miał być" są kluczowe, ponieważ Bajtek doznał szoku kulturowego. Okazało się, że zamiast ulubionych gier Bajtka takich jak rzucanie piłeczką w puszkę, czy łowienie kaczek, w Bitockich parkach rozrywki gra się w zgadywanie ciągów.

Bajtek pracowicie robił zadanka cały rok i nie chce się przemęczać na wakacjach. To powiedziawszy, nie zamierza też wrócić do domu z pustymi rękami. Dlatego to właśnie Ty musisz pomóc Bajtkowi wygrać.

Zasady gry są następujące: Na początku wybierany jest ciąg **parami różnych** liczb, długości n , który trzeba odgadnąć. W tym celu można używać dwóch operacji:

- 1 Pytanie "Jaka jest i -ta liczba w ciągu?"
- 2 Pytanie "Jaki jest zbiór xorów¹ wszystkich par² liczb z danego podciągu?"

Na przykład dla ciągu $[2, 1, 3, 7]$ odpowiedzią na pytanie 1 dla $i = 4$ jest 7, a odpowiedzią na pytanie 2 dla podciągu o indeksach $[2, 3]$ (czyli $[1, 3]$) jest multizbiór $\{0, 0, 2, 2\}$, ponieważ:

- $1 \oplus 1 = 0$
- $1 \oplus 3 = 2$
- $3 \oplus 1 = 2$
- $3 \oplus 3 = 0$

Bajtek może zadać co najwyżej 100 pytań, ale im mniej pytań zada, tym większa będzie nagroda! Po zadaniu wszystkich pytań, trzeba podać cały ciąg. Jeżeli wszystkie liczby będą się zgadzały, Bajtek wygra.

Interakcja

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n ($1 \leq n \leq 100$) oznaczająca długość ciągu. Następnie możesz wykonywać następujące operacje:

- Wypisz "1 i " i wczytaj jedną liczbę całkowitą a będącą odpowiedzią na pytanie 1.
- Wypisz "2 k " ($1 \leq k \leq n$) oraz k liczb x_i będących indeksami podciągu o który chcesz zapytać. W odpowiedzi otrzymasz k^2 liczb będących zbiorem xorów par liczb z podciągu w kolejności niemalejącej.
- Wypisz "3" oraz n liczb będących ciągiem, który chcesz zgadnąć.

Wymagania:

- Twój program nie może zadać więcej niż 100 pytań (operacja 3 nie jest pytaniem).
- Twój program musi wykonać operację 3 dokładnie raz, jako ostatnią.
- Wszystkie indeksy w pytaniach muszą być liczbami całkowitymi z przedziału od 1 do n .
- Indeksy w jednym pytaniu 2 muszą być parami różne.
- Liczba indeksów w jednym pytaniu 2 nie może przekraczać n .

Możesz założyć że w ciągu nie ma dwóch takich samych liczb oraz że wszystkie liczby w ciągu są z przedziału od 0 do 10^9 . Pamiętaj, aby opróżnić bufor wyjściowy przy wysyłaniu komunikatu. To znaczy, jeśli chcesz wysłać jako komunikat pytanie "1 5", zrób to następująco:

- Dla C++: `cout << "1 5" << endl; lub printf("1 5\n"); fflush(stdout);`
- Dla Pythona: `print("1 5", flush=True)`

W zakładce Pliki znajduje się przykładowy program, który nie jest rozwiązaniem, ale pokazuje, w jaki sposób może przebiegać komunikacja.

¹xor bitowy https://pl.wikipedia.org/wiki/Operacje_bitowe
²"Każdy z każdym"

Przykłady

Wejście dla testu r2e0:

```
4
2
3
0 0 6 6
0 0 0 1 1 4 4 5 5
7
```

Wyjście dla testu r2e0:

```
1 1
1 3
2 2 2 4
2 3 1 3 4
1 4
3 2 1 3 7
```

Wyjaśnienie: Początkowa liczba 4 jest długością szukanego ciągu. Najpierw program zapytał się o pierwszy element ciągu, i dostał odpowiedź że jest to 2. Potem o 3ci i dostał odpowiedź że jest to 3. Następnie zadał 2 pytania o zbiory par xorów, najpierw ze zbioru drugiego oraz czwartego elementu, a potem ze zbioru pierwszego trzeciego i czwartego. Potem zadał jeszcze jedno pytanie o element czwarty i odpowiedział zgadując ciąg $[2, 1, 3, 7]$. Jest to poprawna odpowiedź, program spełnił wszystkie wymagania oraz zadał co najwyżej 15 pytań, więc za ten test dostałby 100 punktów.

Ocenianie

Jeżeli końcowy ciąg nie będzie się zgadzał z wylosowanym lub nie spełnisz wymienionych wymagań, Twój program dostanie 0 punktów. W przeciwnym wypadku:

- Jeżeli Twój program zadał co najwyżej 15 pytań, dostaniesz 100% punktów za ten test.
- Jeżeli Twój program zadał więcej niż 15 pytań, dostaniesz $(101 - p)\%$ punktów za ten test, gdzie p to liczba pytań.

Podzadanie	Ograniczenia	Limit czasu	Punkty
1	Maszyna losująca się zepsuła i zawsze losuje ciąg $[1, 2, 3, \dots, n]$	1 s	3
2	brak dodatkowych ograniczeń	1 s	97