

Zadanie: KUL

Kulki

polish

ONTAK, dzień 4. Dostępna pamięć: 256 MB. Limit czasu: 2 s.

30.06.2025

Dane jest N kulek ponumerowanych od 0 do $N - 1$, z których jedna lub dwie kulki są wyróżnione. Twoim zadaniem jest znaleźć numery wyróżnionych kulek. Żeby się tego dowiedzieć, możesz zadawać zapytania – co najwyżej N . Każde zapytanie to t zbiorów $A_1, \dots, A_t$, gdzie $1 \leq t \leq 10$, a zbiory A_i są podzbiorami zbioru $\{0, 1, 2, \dots, N - 1\}$. Jako odpowiedź otrzymujesz dla każdego zbioru A_i informację, czy zawiera on jakiś wyróżniony element. Można zadać co najwyżej N zapytań. We wszystkich testach poza przykładowymi **wartość N jest równa 1024**.

W przeciwieństwie do innych zadań interaktywnych, tym razem przy punktacji nie będzie brana pod uwagę liczba zapytań. Zamiast tego, każde zapytanie ma koszt równy

$$\frac{N}{\min_i(|A_i|)}.$$

Ocena będzie zależeć od łącznego kosztu zapytań, jakie wykona Twój program. Szczegóły dotyczące liczby punktów w zależności od sumy kosztów zapytań znajdują się w sekcji *Ocenianie*.

Interakcja

To jest zadanie interaktywne. Twój program powinien komunikować się z programem sprawdzającym, zadawać mu pytania (opisanego powyżej rodzaju), odczytywać odpowiedzi, a na końcu podać numery wyróżnionych kulek. Komunikacja następuje poprzez standardowe wejście i wyjście, w następujący sposób:

1. Najpierw wczytujesz ze standardowego wejścia dwie liczby całkowite N (liczba kulek) oraz K (liczba wyróżnionych kulek).
2. Aby zadać zapytanie o pewną rodzinę zbiorów, Twój program powinien wypisać na standardowe wyjście kolejno następujące wiersze:

```
1 t
s1 A1[1] ... A1[s1]
:
st At[1] ... At[st]
```

Jest to zapytanie o zbiory $A_1, A_2, \dots, A_t$, gdzie zbiór numer j liczy sobie s_j elementów oznaczonych $A_t[1], A_t[2], \dots, A_t[s_t]$. Wartość t powinna być równa **co najwyżej 10**. Elementy w konkretnym zbiorze **nie mogą się powtarzać, ale jeden element może należeć do kilku zbiorów**. Zbiory nie mogą być puste.

3. Na takie pytanie sprawdzarka odpowiada w następującym formacie:

```
b1
:
bt
```

gdzie $b_i = 1$, jeśli w zbiorze A_i znajduje się jakiś wyróżniony element, zaś $b_i = 0$ w przeciwnym przypadku.

4. Po zadaniu wszystkich zapytań należy podać odpowiedź w formacie:

- dla $k = 1$:

2 a,

gdzie a to indeks wyróżnionej kulki;

- dla $k = 2$:

2 a b,

gdzie $a < b$ to indeksy wyróżnionych kulek.

Po każdym zapytaniu i po każdej odpowiedzi należy wywołać polecenie `cout.flush()`, lub `fflush(stdout)` jeżeli używasz `printf`. Należy przestrzegać podanego wyżej formatu wejścia i wyjścia, łącznie z białymi znakami i podziałem na wiersze – w przeciwnym przypadku program może zakończyć się werdyktem *Przekroczenie limitu czasu* zamiast *Błędna odpowiedź*.

Program oceniający (sprawdzarka) nie jest adaptacyjny, czyli wyróżnione kulki są znane od początku, a nie ustalane w zależności od odpowiedzi użytkownika. Dodatkowo, możesz założyć, że testy zostały wygenerowane losowo.

Przykładowa interakcja:

Interaktor	Program	Wyjaśnienie
4 1		$N = 4, K = 1$
	1 3 2 0 2 2 1 2 3 0 2 3	zapytanie o 3 zbiory: $A_1 = \{0, 2\}$ $A_2 = \{1, 2\}$ $A_3 = \{0, 2, 3\}$
0 1 0		możemy wywnioskować, że 1 jest jedyną wyróżnioną kulką
	2 1	odpowiedź to 1, natomiast suma kosztów to $\frac{4}{2} = 2$

Testy przykładowe:

- 0a: $N = 4, K = 1$, kulka 2 jest specjalna
- 0b: $N = 8, K = 2$, kulki 1 oraz 4 są specjalne
- 0c: $N = 1024, K = 1$, kulka 120 jest specjalna
- 0d: $N = 1024, K = 2$, kulki 507 oraz 568 są specjalne

Ocenianie

Dla każdego podzadania podana jest liczba punktów P oraz oczekiwana wartość kosztu S_{opt} . Jeżeli Twój program osiąga (maksymalnie wśród wszystkich testów tego podzadania) sumę kosztów S , to otrzymuje:

- jeśli $S > N^2$, 0 punktów;
- jeśli $N^2 \geq S > N$, $P \cdot 0.12$ punktów;
- jeśli $N \geq S \geq S_{opt}$, $P \cdot \left(0.12 + 0.88 \cdot \left(1 - \frac{\log(S) - \log(S_{opt})}{\log(N) - \log(S_{opt})}\right)^4\right)$ punktów;
- jeśli $S_{opt} \geq S$, P punktów.

W szczególności oznacza to 100% punktów za $S \leq S_{opt}$ a około 64% punktów gdy $K = 2$ i $S = 2 \cdot S_{opt} = 8$.

Podzadanie	Ograniczenia	Punkty
1	$K = 1, S_{opt} = 2$	23
2	$K = 2, S_{opt} = 4$	77

Eksperymenty

W zakładce *Pliki* możesz pobrać przykładową sprawdzarkę. Wywołuje się ją komendą

```
python3 run.py 1 kulsoc [rozwiązanie] [test]
```

przy czym plik `kulsoc.cpp` oraz rozwiązanie muszą być skompilowane. Na przykład:

```
python3 run.py 1 kulsoc kul kul0b.in
```