Problema Game

C header	`game.h`
C++ header	`game.h`

După un an de jucat hide-and-seek, Lulu și Tanaka în sfârșit s-au găsit. Acum este momentul ca ei să se întoarcă la rezolvarea problemelor imposibile de programare.

Astăzi, Lulu a venit la Tanaka cu încă un joc algoritmic interesant, care este jucat în două runde:

Runda 1 Fiind date două numere N și K, Tanaka trebuie să aleagă o mulțime de intervale $[x, y]$. Pentru fiecare astfel de interval, acesta trebuie să verifice $1 \leq x \leq y \leq N$. Cardinalul mulțimii trebuie să fie maxim 200000.

Runda 2 Lulu îi pune lui Tanaka mai multe întrebări sub formă de intervale $[a, b]$ cu $1 \leq a \leq b \leq N$. Pentru fiecare query, Tanaka trebuie să aleagă o submulțime $[x_1, y_1], [x_2, y_2], \ldots, [x_p, y_p]$ de intervale din Runda 1, astfel încât pentru toate valorile *val* conținute de $[a, b]$, există cel puțin un interval în submulțime care conține *val*, și pentru toate valorile *val* care nu sunt conținute de $[a, b]$, nu există niciun interval din submulțime care să conțină *val*. **În plus, trebuie ca $p \leq K$ pentru fiecare query.** (Spunem că un număr *val* este conținut de un interval $[x, y]$ dacă și numai dacă $x \leq val \leq y$).

Din moment ce lui Lulu îi plac intervalele disjuncte, **el îi va da lui Tanaka** 100% **din punctajul jocului doar dacă pentru fiecare query, toate intervalele alese de Tanaka sunt disjuncte.** Totuși, dacă toate răspunsurile la întrebări sunt corecte, dar intervalele nu sunt disjuncte, Tanaka va primi 50% din punctaj.

De asemenea, Lulu vrea ca mulțimea aleasă în runda 1 să fie destul de mică. După ce a auzit această problemă, Tanaka i-a spus lui Lulu: "Nu va fi greu să răspund la întrebările tale, Lulu!" Totuși, are nevoie de ajutorul tău. Cerința ta este să scrii un program care alege mulțimea inițială și după răspunde la toate întrebările lui Lulu.

Protocolul de interacțiune

Concurentul trebuie să implementeze două funcții:

```
void init(int N, int K);
void query(int a, int b);
```

Concurentul poate apela următoarea funcție:

```
void chooseInterval(int left, int right);
```

Funcția `init` va fi apelată **o singură dată**, la începutul interacțiunii. Funcția va primi ca parametri valorile N și K, cele două numere primite de Tanaka în Runda 1. Concurentul trebuie să apeleze funcția `chooseInterval` de cel mult 200000 de ori cu intervalele pe care Tanaka le alesese inițial în Runda 1. După, comisia va apela funcția `query` de mai multe ori. Va primi ca parametri valorile a și b, reprezentând un query. Concurentul trebuie să apeleze din nou funcția `chooseInterval` pentru intervalele alese de Tanaka din lista furnizată în Runda 1.

Atenție! Concurentul nu trebuie să implementeze funcția main si trebuie să includă `#include` header-ul `game.h` ! Concurenților le este permisă folosirea variabilelor globale și a altor funcții, care nu vor fi resetate între interacțiuni.

Punctare

Fie C numărul de intervale care vor fi alese de program și M numărul maxim de intervale care pot fi alese pentru fiecare subtask. Fie *Points* scorul maxim pentru un subtask. Fie *disjoint* = 1 dacă intervalele

alese de Tanaka pentru fiecare query sunt disjuncte, și $disjoint = \frac{1}{2}$ altfel. Dacă $C > 2 \times 10^5$, atunci scorul pentru test va fi 0. Altfel, scorul pentru fiecare test din subtask va fi egal cu:

$$S = Points \times \min\left(1, \frac{M}{C}\right) \times disjoint$$

Punctajul pentru un subtask va fi scorul minim S dintre scorurile tuturor testelor sale.

Restricții

- Fie Q numarul de apelări ale funcției `query` de către comisie.

#	Puncte	N	K	Q	M
1	6	50	1	2500	1275
2	6	1000	2	5000	7997
3	9	10000	2	50000	113645
4	9	1000	3	5000	5485
5	11	10000	3	50000	76989
6	8	10000	4	50000	58962
7	7	10000	5	50000	47986
8	6	10000	6	50000	40956
9	6	10000	7	50000	36011
10	6	10000	8	50000	33911
11	7	10000	9	50000	30923
12	9	10000	10	50000	26598
13	10	1000	20	50000	1786

Exemple

Date de Intrare	Date de Ieșire
`init(5,3)`	

	`chooseInterval(1,1)`
	`chooseInterval(2,3)`
	`chooseInterval(4,4)`
	`chooseInterval(4,5)`

`query(1,3)`	
	`chooseInterval(1,1)` `chooseInterval(2,3)`
`query(2,4)`	
	`chooseInterval(2,3)` `chooseInterval(4,4)`
`query(1,5)`	

	`chooseInterval(1,1)`
	`chooseInterval(2,3)`
	`chooseInterval(4,5)`

Explicații

Funcția `init` va fi apelată de comisie, cu valorile $N = 5$ și $K = 3$. Intervalele alese sunt: $[1, 1], [2, 3], [4, 4], [4, 5]$.

Pentru prima interogare, $a = 1$ și $b = 3$. Intervalele alese sunt: $[1, 1], [2, 3]$. Numărul 1 este conținut de $[1, 1]$, 2 este conținut $[2, 3]$, iar 3 este conținut de $[2, 3]$.

Pentru a doua interogare, $a = 2$ și $b = 4$. Intervalele alese sunt: $[2, 3], [4, 4]$. Numărul 2 este conținut de $[2, 3]$, 3 este conținut de $[2, 3]$, iar 4 este conținut de $[4, 4]$.

Pentru a treia interogare, $a = 1$ și $b = 5$. Intervalele alese sunt: $[1, 1], [2, 3], [4, 5]$. Numărul 1 este conținut de $[1, 1]$, 2 este conținut de $[2, 3]$, 3 este conținut de $[2, 3]$, 4 este conținut de $[4, 5]$ iar 5, este conținut de $[4, 5]$.