

## Задача Floppy

C header: floppy\_c.h  
C++ header: floppy.h

Роксет попаднала на масив  $v_0, v_1, v_2, \dots, v_{N-1}$  от  $N$  **различни** цели числа. Този масив бил много интересен за нея, затова тя искала да го запише на дискета. Понеже там имало малко свободно място, Роксет била принудена да запише нещо по-малко, без да има възможност да запише целия масив. Тя планира да запише само масив от битове, който ще ѝ позволи да отговаря на всяка заявка от следния тип:

$query(a, b) = id$ , където  $a \leq id \leq b$  и  $v_{id} = \max(v_a, v_{a+1}, \dots, v_{b-1}, v_b)$

С други думи, заявката връща индекса на максималната стойност в запитания подмасив.

Роксет иска вашата помощ. Два пъти! Първо, тя ще Ви предостави интересния масив и Вие ще трябва да ѝ кажете каква поредица от битове да запази на дискетата си. Второ, ако тя има нужда да знае отговора на някакви заявки, ще Ви даде редицата от битове, която сте ѝ казали по-рано, и заявките, на които се нуждае от отговор. Вие трябва да ѝ върнете правилните отговори за подадените заявки.

### Детайли по имплементацията

**Тази задача е тип комуникация - има две интеракции!**

Състезателят трябва да имплементира две функции. Първата от тях трябва да има следната сигнатура:

```
(C) void read_array(int subtask_id, int N, int* v);  
(C++) void read_array(int subtask_id, const std::vector<int> &v);
```

Тази функция ще бъде извикана **точно веднъж**, за първата интеракция, и тя ще предостави на състезателя интересния масив на Роксет. В имплементацията на тази функция, състезателят трябва да извика следната функция **точно веднъж**, която ще е имплементирана от grader-a:

```
(C) void save_to_floppy(int L, char* bits);  
(C++) void save_to_floppy(const std::string &bits);
```

Тази функция казва на Роксет каква поредица от битове да запише на нейната дискета. Параметърът  $L$  е броят битове, които да бъдат запазени. Параметърът  $bits$  трябва да е поредица от символи (само символите '0' и '1' са разрешени).

Втората функция, която трябва да имплементират състезателите трябва да има следната сигнатура:

```
(C) int* solve_queries(int subtask_id,  
                     int N, int L, char* bits,  
                     int M, int* a, int* b);  
(C++) std::vector<int> solve_queries(int subtask_id,  
                                     int N, const std::string &bits,  
                                     const std::vector<int> &a,  
                                     const std::vector<int> &b);
```

Тази функция ще бъде извикана **точно веднъж**, за втората интеракция, и ще предостави на състезателя дължината на интересния масив на Роксет, масивът с битове, който състезателя е дал по-рано на Роксет за записване на дискетата, и списък от  $M$  заявки.

Параметрите на  $i$ -тата заявка са  $a[i]$  и  $b[i]$ .

Тази функция трябва да върне масив от  $M$  цели числа - съответния отговор на всяка заявка ( $i$ -тото цяло число трябва да бъде отговора на  $i$ -тата заявка).

**Важно:** Програмата на състезателя ще бъде пусната два пъти, по един път за всяка интеракция. Затова, всякакви пресметнати данни от програмата на състезателя по време на първото пускане, няма да бъдат достъпни по време на второто пускане.

## Ограничения

- $-10^9 \leq v_i \leq 10^9$  за всяко  $0 \leq i \leq N - 1$
- $1 \leq L \leq 200\,000$

## Подзадача 1 (7 точки)

- $1 \leq N \leq 500$
- $0 \leq v_i < N$  за всяко  $0 \leq i \leq N - 1$
- $1 \leq M \leq 1\,000$
- Резултатът за подзадачата ще бъде 7, ако всички тестове са решени правилно и ще бъде 0, иначе.

## Подзадача 2 (21 точки)

- $1 \leq N \leq 10\,000$
- $1 \leq M \leq 20\,000$
- За всеки правилно решен тест резултатът ще е  $\min(1, \frac{1}{2^{\frac{L}{N}-1-\log_2 N}})$ . Резултатът за тази подзадача ще бъде минимума от резултатите на всеки тест, умножен по 21.

## Подзадача 3 (72 точки)

- $1 \leq N \leq 40\,000$
- $1 \leq M \leq 80\,000$
- За всеки правилно решен тест резултатът ще е  $\min(1, \frac{1}{2^{\frac{L}{N}-2}})$ . Резултатът за тази подзадача ще бъде минимума от резултатите на всеки тест, умножен по 72.

## Примерна интеракция

За първата интеракция, функция на състезателя ще бъде извикана така:

```
read_array(  
    /* subtask_id = */ 3,  
    /* v          = */ {40, 20, 30, 10});
```

Тази функция може да реши да запази следните битове на дискетата:

```
save_to_floppy(  
    /* bits = */ "001100");
```

Първото пускане на програмата на състезателя ще бъде прекратено, т.е. всякакви данни, които са били записани в паметта от програмата, ще бъдат загубени.

За втората интеракция, функция на състезателя ще бъде извикана така:

```
solve_queries(  
    /* subtask_id = */ 3,  
    /* N          = */ 4,  
    /* bits       = */ "001100",  
    /* a          = */ {0, 0, 0, 0, 1, 1, 1, 2, 2, 3},  
    /* b          = */ {0, 1, 2, 3, 1, 2, 3, 2, 3, 3});
```

Тази функция трябва да върне масив от  $M$  цели числа:

```
{0, 0, 0, 0, 1, 2, 2, 2, 2, 3}
```