

## Problem Floppy

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

Технарка Роксета наткнулась на масив  $v_0, v_1, v_2, \dots, v_{N-1}$  з  $N$  **різних** цілих чисел. Вона зрозуміла, що це дуже цікавий масив, тому захотіла зберегти його на своєму диску. Однак через те, що вільного місця на диску мало, Роксета має задовольнятися меншим: вона не зможе зберегти весь масив, натомість вона планує зберегти масив бітів, що дасть їй змогу відповідати на запити виду:

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

Іншими словами, запит має повернути позицію, на якій міститься максимальний елемент на заданому відрізку.

Роксета просить Вас про допомогу. Двічі! По-перше, вона надасть вам цей цікавий масив, і Вам потрібно сказати їй, яку послідовність бітів потрібно зберегти на її диску. По-друге, якщо їй потрібно знати відповідь на деякі запити, вона надасть Вам послідовність бітів, які ви сказали їй зберегти на диску, і запити, на які їй потрібно відповісти, тоді як Ви повинні дати їй правильну відповідь на кожен із запитів.

### Деталі взаємодії

Ця задача складається з двох інтерактивних!

Ви маєте реалізувати дві функції. Перша з них нижче:

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

Ця функція буде викликана **рівно один раз**, під час першої взаємодії з Вашою програмою, і забезпечує Вас масивом, що зацікавив Роксету. В реалізації цієї функції, Ви маєте викликати наступну функцію **рівно один раз**, яка буде реалізована програмою журі:

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

Ця функція каже Роксеті, яку послідовність бітів вона має зберегти на свій диск. Параметр  $L$  позначає кількість бітів, що мають бути збережені. Параметр  $bits$  має бути послідовністю символів (дозволені тільки символи '0' та '1').

Друга функція, яку Ви маєте реалізувати:

```
(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);
```

Ця функція буде викликана **рівно один раз**, під час другої взаємодії, і Ви отримаєте довжину масиву, що зацікавив Роксету, масив бітів, який Роксета зберегла на диску та список з  $M$  запитів.

Параметри  $i$ -го запиту такі:  $a[i]$  та  $b[i]$ .

Ця функція має повертати масив з  $M$  цілих чисел, що означають відповіді до кожного з запитів ( $i$ -е число має бути відповіддю до  $i$ -го запиту).

**Важливо:** Програма учасника буде запускатись двічі, по одному разу на кожний вид взаємодії. Отже, будь-які дані, обчислені програмою учасника під час першої взаємодії не будуть доступні під час другої взаємодії.

## Обмеження

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

## Підзадача 1 (7 балів)

- $1 \leq N \leq 500$
- $0 \leq v_i < N$  для усіх  $0 \leq i \leq N - 1$
- $1 \leq M \leq 1\,000$
- Ви отримаєте за цю підзадачу 7 балів, якщо усі тести будуть правильно вирішені, інакше ви отримаєте 0 балів.

## Підзадача 2 (21 балів)

- $1 \leq N \leq 10\,000$
- $1 \leq M \leq 20\,000$
- Бал за кожен тест дорівнює  $\min(1, \frac{1}{2^{\frac{L}{N}-1-\log_2 N}})$ . Сумарний бал за підзадачу дорівнює мінімуму з балів, що ви отримали за кожен тест, і помножений на 21.

## Підзадача 3 (72 балів)

- $1 \leq N \leq 40\,000$
- $1 \leq M \leq 80\,000$
- Бал за кожен тест дорівнює  $\min(1, \frac{1}{2^{\frac{L}{N}-2}})$ . Сумарний бал за підзадачу дорівнює мінімуму з балів, що ви отримали за кожен тест, і помножений на 72.

## Приклад взаємодії

Під час першої взаємодії буде визвана функція учасника:

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

Ця функція, наприклад, обирає для збереження наступні біти:

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

Першу роботу програми учасника буде припинено, отже, будь-які дані, що зберігає програма будуть втрачені.

Під час другої взаємодії, буде визвана функція учасника:

```
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});
```

Ця функція має повернути масив з  $M$  чисел:

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