

## Танцуващи слонове

*Танцуващи слонове* е впечатляващо шоу, представяно в Патая –  $N$  слона танцуват в редица, която ще наричаме *сцена*.

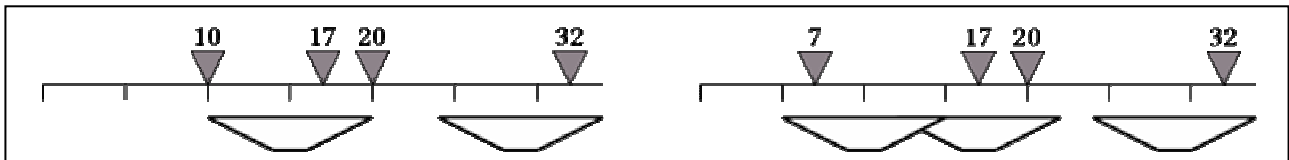
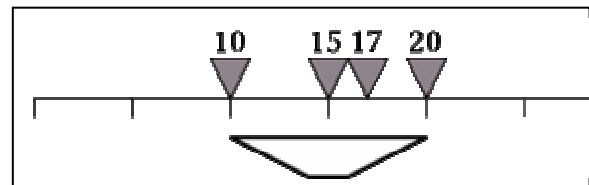
След години тренировки, слоновете са научили прекрасни танци, съставени от отделни действия. Във всяко действие един слон напуска танцувайки позицията, в която се намира и се премества на друга.

Продуцентите на шоуто искат да направят фото-албум показващ цялото шоу, като след всяко действие заснемат всички слонове така, както ги виждат зрителите.

Във всеки момент от шоуто, на една и съща позиция могат да се намират по няколко слона и тогава те застават един зад друг на тази позиция.

С една камера може да се снима група от слонове, тогава и само тогава, когато позициите им се намират във вътрешността или в краищата на отсечка с дължина  $L$ . Тъй като слоновете може да са пръснати по цялата сцена, може да се наложи да се използват няколко камери, за да се снимат слоновете след поредното действие.

Например, нека  $L=10$  и слоновете са на позиции 10, 15, 17 и 20. В този момент е достатъчна една камера (вж. Фигурата вдясно, където слоновете са показани като триъгълници, а камерите – като трапеци).



Ако слонът от 15-та позиция се премести в 32 (вж. Фигурата горе вляво), ще са нужни две камери за да снимат сцената. Ако сега слонът от 10-та позиция се премести в 7 (вж. Фигурата горе вдясно), ще са нужни три камери за да снимат сцената.

В тази интерактивна задача, ще трябва да определите *минималният* брой камери, необходими за заснемането на сцената след всяко действие. Този брой може да се увеличава, спрямо броя на камерите, необходими за предишната сцена, да намалява или да остава същия.

### Задача

Напишете:

❖ функция  $\text{init}(N, L, X)$  със следните параметри:

- $N$  – броят на слоновете, номерирани от  $0$  до  $N - 1$ .
- $L$  – дължина на отсечката, заснимана от една камера,  $L$  е цяло,  $0 \leq L \leq 1\,000\,000\,000$ .
- $X$  – едномерен масив от цели, задаващ началните позиции на слоновете. За  $0 \leq i < N$ , началната позиция на слона с номер  $i$  е в  $X[i]$ . Началните позиции са сортирани:  $0 \leq X[0] \leq X[1] \leq \dots \leq X[N - 1] \leq 1\,000\,000\,000$ . По време на танца, слоновете могат да променят позициите си един спрямо друг.

Тази функция не връща стойност и ще бъде извикана само веднъж, преди всички извиквания на функцията **update**.

❖ Функция **update(i,y)** със следните параметри:

- **i** – номерът на слона, който сменя позицията си в поредното действие.
- **y** – позицията, на която слонът с номер **i** ще застане в края на действието, **y** е цяло,  $0 \leq y \leq 1\,000\,000\,000$ .

Функцията ще бъде извиквана многократно – по едно извикване за всяко действие (и то трябва да бъде приложено към сцената, получена при предното действие). При всяко извикване, тя трябва да върне *минималния брой камери, които са необходими за да се снимат всички слонове след поредното действие*.

## Пример

Нека вашата функция **init** е извикана еднократно с параметри **N=4**, **L=10**, **X={10,15,17,20}**, след което многократно, по един път за всяко действие, се извиква вашата функция **update**. В таблицата е показана примерна последователност от действия – параметрите които функцията получава и стойностите които трябва да върне.

№	извикване	връща
1	update(2,16)	1
2	update(1,25)	2
3	update(3,35)	2
4	update(0,38)	2
5	update(2,0)	3

## Подзадачи

### Подзадача 1 (10 т.)

- Слоновете са само 2
- В началото и след всяко действие, слонове ще бъдат на различни позиции.
- Функцията **update** ще бъде извикана най-много 100 пъти.

### Подзадача 2 (16 т.)

- $1 \leq N \leq 100$
- В началото и след всяко действие, всички слонове ще бъдат на различни позиции.
- Функцията **update** ще бъде извикана най-много 100 пъти.

### Подзадача 3 (24 т.)

- $1 \leq N \leq 50\,000$
- В началото и след всяко действие, всички слонове ще бъдат на различни позиции.
- Функцията **update** ще бъде извикана най-много 50 000 пъти.

### Подзадача 4 (47 т.)

- $1 \leq N \leq 70\,000$
- Няколко слона могат да бъдат на една позиция.
- Функцията **update** ще бъде извикана най-много 70 000 пъти.

### Подзадача 5 (3 т.)

- $1 \leq N \leq 150\,000$
- Няколко слона могат да бъдат на една позиция.
- Функцията **update** ще бъде извикана най-много 150 000 пъти.

## Детайли на реализацията

### Ограничения

- По време: 9 сек.  
**Забележка.** Темплейтите на стандартната библиотека на C++ (STL) могат да се окажат бавни; в частност, може да не успеете да решите подзадача 5, ако ги използвате.
- По памет: 256 МВ.  
**Забележка:** Няма ограничение на размера на стека в явен вид. Паметта отделена за стек е част от общата достъпна памет.

### Интерфейс (API)

- Папка на задачата: `elephants/`
- Файл с решение на състезателя: `elephants.c` или `elephants.cpp` или `elephants.pas`
- Интерфейсен файл на състезателя: `elephants.h` или `elephants.pas`
- Примерен грейдър: `grader.c` или `grader.cpp` или `grader.pas`
- Примерни входни файлове на грейдъра: `grader.in.1`, `grader.in.2`, ...  
**Забележка:** Примерният грейдър чете входни файлове със следния формат:
  - Ред 1:  $N$ ,  $L$  и  $M$ , където  $M$  е броят на действията в шоута.
  - Редове от 2 до  $N + 1$ : началните позиции, за  $0 \leq k < N$ , редът с номер  $k + 2$  съдържа  $X[k]$ .
  - Редове от  $N + 2$  до  $N + M + 2$ : данните за  $M$ -те действия, редът с номер  $N + 1 + j$  съдържа  $i[j]$ ,  $y[j]$  и  $s[j]$ , разделени с по един интервал, което значи, че при  $j$ -тото действие слонът с номер  $i[j]$  се мести на позиция  $y[j]$  и след това, минималният брой на необходимите камери е  $s[j]$ ,  $1 \leq j \leq M$ .
- Файлове с очакваните резултати за примерния грейдър: `grader.expect.1`, `grader.expect.2`, ... Всеки от тези файлове трябва да съдържа точно текста "Correct."