



## Zid

Jian-Jia (Janez) gradi zid, tako da zlaga enako velike zidake enega na drugega. Zid sestavlja  $n$  stolpcev zidakov, ki so z leve proti desni oštevilčeni od 0 do  $n - 1$ .

Stolpci so lahko različnih višin. Višino stolpca predstavlja število zidakov v njem.

Jian-Jia gradi zid, kot sledi. Na začetku v nobnem stolpcu ni nobenega zidaka. Nato gre Jian-Jia skozi  $k$  korakov gradnje, kjer je en korak bodisi korak *dodajanja* bodisi korak *odvzemanja* zidakov.

Gradnja je zaključena, ko je zaključenih vseh  $k$  korakov.

Na vsakem koraku Jian-Jia dobi obseg zaporednih stolpcev zidakov in višino  $h$ , izvede pa naslednji postopek:

- V koraku *dodajanja*, Jian-Jia dodaja zidake tistim stolpcem v podanem obsegu, ki imajo manj kot  $h$  zidakov, tako da imajo ti stolpci natanko  $h$  zidakov. Pri stolpcih s  $h$  ali več zidaki ne naredi ničesar.
- V koraku *odstranjevanja*, Jian-Jia odstranjuje zidake tistim stolpcem v podanem obsegu, ki imajo več kot  $h$  zidakov, tako da imajo ti stolpci natanko  $h$  zidakov. Pri stolpcih s  $h$  ali manj zidaki ne naredi ničesar.

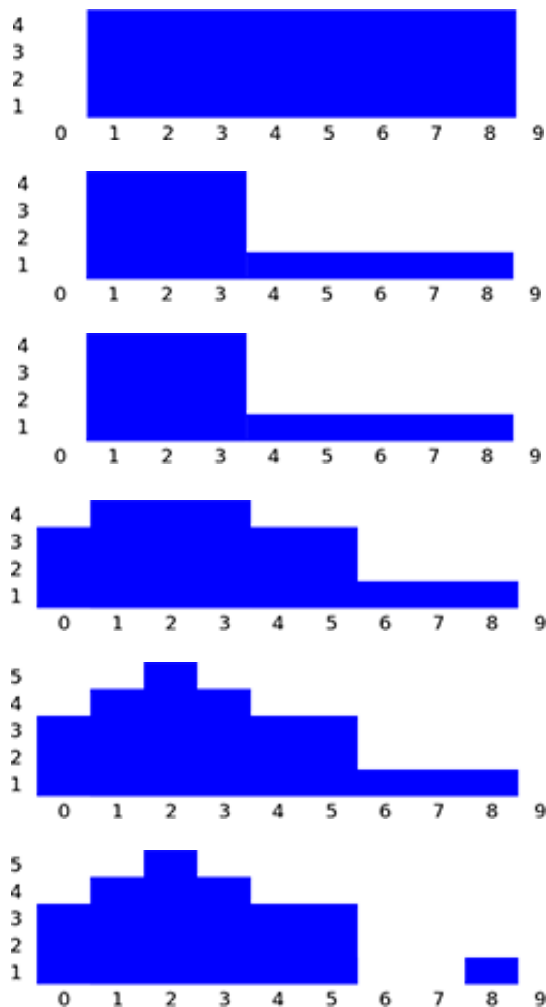
Tvoja naloga je ugotoviti končno obliko zidu.

## Primer

Predpostavimo da imamo 10 stolpcev zidakov in 6 korakov gradnje. Podani obsegi vedno vsebujejo tudi robne stolpce. Diagrami stanja zidu, po vsakem koraku gradnje, so prikazani spodaj.

korak	tip	obseg	višina
0	dodajanje	stolpci 1 do 8	4
1	odstranjevanje	stolpci 4 do 9	1
2	odstranjevanje	stolpci 3 do 6	5
3	dodajanje	stolpci 0 do 5	3
4	dodajanje	stolpec 2	5
5	odstranjevanje	stolpci 6 do 7	0

Sprva so vsi stolpci prazni, zato bodo po koraku 0 stolpci od 1 do 8 vsebovali po 4 zidake; stolpec 0 in 9 ostaneta prazna. V koraku 1 se odstranjuje zidake iz stolpcev od 4 do 8, vse dokler niso vsi stolpci višine 1. Stolpec 9 ostaja prazen, stolpci od 0 do 3 so izven podanega obsega, zato ostanejo nespremenjeni. Korak 2 ne povzroči sprememb, saj stolpci od 3 do 6 ne vsebujejo več kot 5 zidakov. Po koraku 3 se število zidakov v stolpcih 0, 4 in 5 poveča na 3. Po koraku 4 je v stolpcu 2 natanko 5 zidakov. Korak 5 odstrani vse zidake iz stolpcev 6 in 7.



## Naloga

S podanimi opisi  $k$  korakov gradnje, izračunaj število zidakov vsakega stolpca po zaključeni gradnji. Implementirati moraš funkcijo `buildWall`.

- `buildWall(n, k, tip, levi, desni, visina, koncneVisine)`
  - $n$ : število stolpcev zidu.
  - $k$ : število korakov gradnje.
  - `tip`: tabela dolžine  $k$ ; `tip[i]` je tip koraka  $i$ : 1 predstavlja korak dodajanja in 2 korak odstranjevanja (za  $0 \leq i \leq k - 1$ ).
  - `levi` in `desni`: tabeli dolžine  $k$ ; obseg stolpcev vključenih v korak  $i$  se prične s stolpcem `levi[i]` in zaključi s stolpcem `desni[i]` (vsebuje `levi[i]` in `desni[i]`), za  $0 \leq i \leq k - 1$ . Vedno velja `levi[i] ≤ desni[i]`.
  - `visina`: tabela dolžine  $k$ ; `visina[i]` je parameter višine pri  $i$ -tem koraku, za  $0 \leq i \leq k - 1$ .
  - `koncneVisine`: tabela dolžine  $n$ ; svojo rešitev vračaš preko te tabele -- končno število zidakov v stolpcu  $i$  zapišeš v `koncneVisine[i]`, za  $0 \leq i \leq n - 1$ .

# Podnaloge

Pri vseh podnalogah velja, da je parameter višine za vse korake nenegativno celo število in je manjše ali enako 100,000.

podnaloge	točke	$n$	$k$	opomba
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	ni dodatnih omejitev
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	vsi koraki dodajanja so pred koraki odstranjevanja
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	ni dodatnih omejitev
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	ni dodatnih omejitev

## Podrobnosti implementacije

Oddati moraš natanko eno datoteko, imenovano `wall.c`, `wall.cpp` ali `wall.pas`. V tej datoteki implementiraj zgoraj opisano funkcijo, z naslednjimi podpisi. V C/C++ morate obvezno vključiti tudi header datoteko `wall.h`.

### V programskem jeziku C/C++

```
void buildWall(int n, int k, int tip[], int levi[], int desni[],  
int visina[], int koncneVisine[]);
```

### V programskem jeziku Pascal

```
procedure buildWall(n, k : longint; tip, levi, desni, visina :  
array of longint; var koncneVisine : array of longint);
```

## Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod v sledečem formatu:

- vrstica 1:  $n, k$ .
- vrstice  $2 + i$  ( $0 \leq i \leq k - 1$ ):  $tip[i], levi[i], desni[i], visina[i]$ .