

Problema Shperl

Fișier de intrare: **shperl.in**
Fișier de ieșire: **shperl.out**

Limită de timp: 0,4 secunde
Limită de memorie: 256 MB

Pentru că lumea are nevoie de mai mulți programatori, un consorțiu a inventat limbajul Shperl. Acesta este atât de simplu, că oricine poate deveni programator, fără ca măcar să meargă la facultate! În Shperl există un singur tip de date, întregul pe N biți, și doar două tipuri de operații:

- **1 X Y**: neagă toți biții aflați pe poziții între X și Y inclusiv;
- **2 X Y**: raportează numărul de biți 1 aflați pe poziții între X și Y inclusiv.

Consortiul vă angajează să testați primul compilator de Shperl din lume, verificând corectitudinea unui șir de Q operații.

Date de intrare

Prima linie conține întregii N și Q , separați prin spațiu. Următoarea linie conține N caractere '0' și '1', **nedespărțite**. Următoarele Q linii conțin câte trei întregi T , X și Y care codifică o operație.

Date de ieșire

Pentru fiecare operație de tip 2, în aceeași ordine ca la intrare, afișați câte o linie cu rezultatul operației.

Restricții

- $3 \leq N \leq 300.000$
- $1 \leq Q \leq 300.000$
- $1 \leq X \leq Y \leq N$ pentru toate operațiile.

#	Puncte	Restricții
1	10	$N \leq 10.000, Q \leq 10.000$
2	30	$N \leq 100.000, Q \leq 100.000$
3	20	$N \leq 150.000, Q \leq 150.000$
4	20	$N \leq 250.000, Q \leq 250.000$
5	20	Fără restricții suplimentare.

Exemplu

shperl.in	shperl.out	explicații
7 4	3	Pe pozițiile 3-6 avem biții 1101.
1011011	2	După prima modificare numărul devine 1100011.
2 3 6		După a doua modificare numărul devine 1101100.
1 2 4		Pe pozițiile 3-7 avem biții 01100.
1 4 7		
2 3 7		