

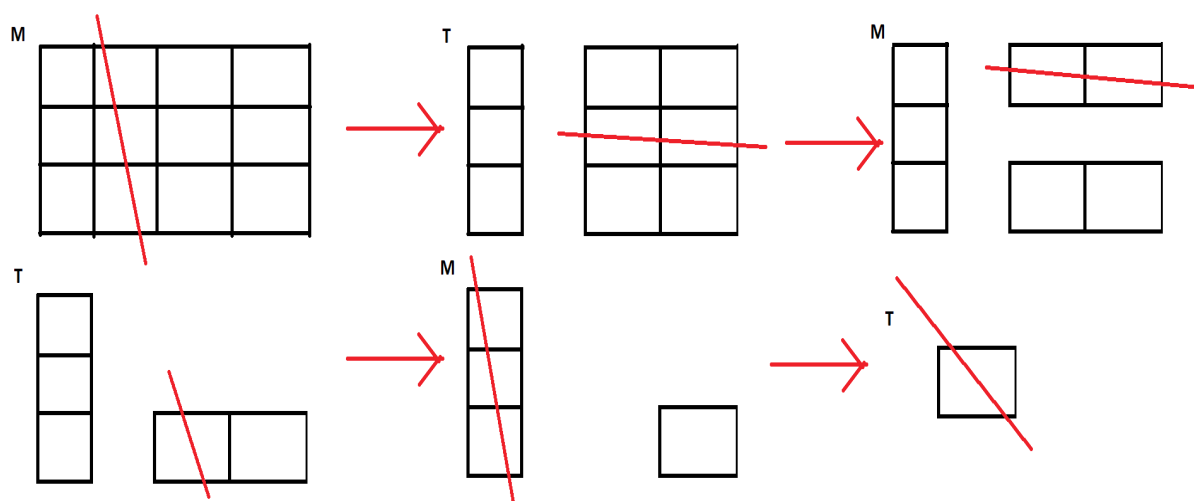
## PROBLEMA 1 MIHAI

100 puncte

Mihai are o temă la compilatoare. Ca să termine tema, trebuie să învingă în lupta cu Tema. Lupta se desfășoară astfel:

Se dă un dreptunghi format din  $N \times M$  pătrate. Mihai începe. La fiecare mutare, jucătorul (Mihai sau Tema) poate alege un rând sau o coloană continuă/continuă nenul/nenulă de pătrate și să o elimine din dreptunghi. Adică pentru o coloană formata din 3 pătrate, se pot elimina fie un pătrat (întrucât se elimina un singur rând continuu), fie toată coloana continuă. În cazul în care un rând care conectează elementele unei coloane a fost eliminat anterior, elementele acelei coloane nu pot fi eliminate dintr-o singură mutare întrucât coloana numai este continuă, ci a fost împărțită în 2 coloane continue (acest aspect este evidențiat în exemplu). Câștigătorul este ultimul care face o mutare.

Acesta este un exemplu de joc unde M este Mihai și T este tema:



În acest exemplu, după ce dreptunghiul este distrus pe parcursul bătăliei, Tema având ultima mutare, câștigă.

**Cerință**

Date fiind  $Q$  dreptunghiuri, Mihai vă roagă să-i spuneți dacă îl bate Tema în scenariul în care atât Mihai cât și Tema joacă optim.

**Date de intrare**

În fișierul *mihai.in* se va afla  $Q$  reprezentând numărul de dreptunghiuri urmate de  $Q$  linii care conțin pe linia  $(i+1)$   $N_i$  și  $M_i$ , dimensiunile celui de-al  $i$ -lea dreptunghi.

**Date de ieșire**

În fișierul *mihai.out* se vor afla  $Q$  linii, pe a  $i$ -a linie fiind rezultatul luptei lui Mihai cu Tema pe un dreptunghi de dimensiuni  $N_i \times M_i$  dacă ambii joacă optim. Rezultatul va fi reprezentat de 1 dacă Mihai câștigă și 0 dacă este inevitabil că Tema să câștige.

## Secțiunea 9-10 avansați

## Restricții și precizări

- $Q \leq 10^6$
- $N, M \leq 10^{18}$
- Pentru 20 % din teste,  $Q = 5$  și  $N * M \leq 10^5$

## Exemple

| mihai.in        | mihai.out | Explicație                            |
|-----------------|-----------|---------------------------------------|
| 1<br>2 1        | 1         | Mihai poate să taie tot rândul direct |
| 2<br>3 4<br>2 2 | 1<br>0    |                                       |

**Timp de executare:** 1.5 secunde/test

**Memorie totală disponibilă:** 16 MB, din care 16 MB pentru stivă

**Dimensiunea maximă a sursei:** 10 KB

## PROBLEMA 2

## RUMBAL

100 puncte

Împăratul Rumbal conducea o împărăție vastă și avea o armată pe măsură. Pentru a se pregăti de război, împăratul Rumbal își antrenează strategiile militare folosind o tablă și soldați în miniatură (pe care îi vom numi Budi). Astăzi, împăratul își dorește să simuleze marșul soldaților, folosind o tablă de forma unui drum în care fiecărei porțiuni din drum îi corespunde câte o literă mare a alfabetului englez, astfel, ne putem imagina tabla ca un vector unidimensional de lungime  $N$ . La începutul exercițiului pe fiecare segment de drum se afla câte un singur Budi, existând în total  $N$  soldați.

Împăratul Rumbal urmează să dea  $Q$  ordine : “Toți Budi care se află pe segmente de drum cu litera  $C$  să se mute un segment la  $D$  (dreapta) sau la  $S$  (stânga).” Dacă un Budi se află pe poziția 1 și primește ordin să se mute la stânga, acesta va cădea de pe tablă. De asemenea, dacă un Budi se află pe poziția  $N$  și primește ordin să se mute la dreapta va cădea de pe tablă. Împăratul Rumbal este bun la dat ordine dar nu și la a anticipa rezultatelor acestora.

## Cerință

Știind literele corespunzătoare tuturor segmentelor și toate ordinele împăratului, spuneți câți Budi vor mai fi rămași pe tablă la final.

## Date de intrare

Pe prima linie a fișierului de intrare *rumbal.in* se vor afla numerele  $N$  și  $Q$  cu semnificațiile din enunț. Pe următoarea linie se va afla un șir de  $N$  litere mari ale alfabetului englez semnificând literele aferente fiecărei căsuțe. Apoi, pe următoarele  $Q$  linii se află câte două caractere, primul reprezentând litera pe care o afectează ordinul și apoi  $S$  sau  $D$  reprezentând direcția.

## Secțiunea 9-10 avansați

## Date de ieșire

Fișierul de ieșire *rumbal.out* va conține un singur număr reprezentând numărul de Budi rămași.

## Restricții și precizări

- $1 \leq N \leq 2 * 10^5$
- $Q \leq 2 * 10^5$

## Exemple

| rumbal.in  | rumbal.out | Explicație   |
|--|------------|--|
| 3 4<br>ABC<br>A S<br>B S<br>B D<br>A D   | 2          | După primul ordin soldatul de pe poziția 1 va cădea de pe tablă. După al doilea ordin soldatul de pe poziția 2 se va muta la stânga pe poziția 1. Ordinul cu numărul 3 nu va avea nici un efect, pe căsuța B nu se afla nici un soldat iar ultimul ordin va muta soldatul de pe poziția 1 înapoi pe poziția 2. |
| 8 3<br>AABCBDDBA<br>A S<br>B D<br>A D  | 5          |  |
| 10 15<br>SNCZWRCEWB<br>B D<br>R D<br>E D<br>W D<br>Z S<br>S D<br>Q S<br>W S<br>B D<br>C S<br>A S<br>N S<br>E D<br>Z S<br>S S | 3          |  |

**Timp maxim de execuție:** 0.1 secunde/test

**Memorie totală disponibilă:** 1 MB din care 1 MB pentru stivă

**Dimensiunea maximă a sursei:** 10 Kb

## Secțiunea 9-10 avansați

## PROBLEMA 3 PEPENI

100 puncte

Într-o zonă puternic radio-activă, Marcel a descoperit o multitudine de pepeni de diverse culori. El știe că aceștia sunt modificați genetic și că dacă îi mănâncă după ce îi tratează poate să capete diverse superputeri (fiecare culoare corespunde unei superputeri). Marcel culege câte  $n$  pepeni din fiecare culoare și îi aduce în garajul său unde a construit un sistem de eliminare a toxinelor din pepeni de lungime  $n$ : a plasat  $q$  neoane de diferite lungimi într-o linie dreaptă care trimit antiradiații (fiecare neon generează  $N_i$  antiradiații) pentru a face pepenii comestibili.

Însă Marcel a întâlnit o problemă. Fiecărui pepene de tipul  $i$  trebuie să îi fie aplicate exact  $P_i$  antiradiații pentru a deveni comestibil. Dacă sunt aplicate mai puține antiradiații, pepenele nu devine comestibil, iar dacă sunt aplicate mai multe antiradiații, pepenele va exploda și nicio superputere nu îl poate salva pe Marcel de o asemenea explozie. În urma multor calcule, Marcel reușește să găsească  $X_i$ , numărul de antiradiații necesare fiecărui tip de pepene pentru a deveni comestibil.

**Cerință**

Marcel își propune să așeze sub neoane, pe rând, fiecare tip de pepeni și să încerce să obțină minim unul comestibil. Știind că odată plasați sub neoane, pepenii nu mai pot fi mutați, Marcel vă roagă să îl ajutați să afle ce puteri poate obține.

**Date de intrare**

Fișierul de intrare *pepeni.in* va conține:

- pe prima linie trei numere:  $n$  (numărul de poziții pe care vor fi plasați pepenii),  $q$  (numărul de neoane construite de Marcel) și  $p$  (numărul de tipuri de pepeni)
- pe următoarele  $q$  linii sunt câte 3 numere care descriu neoanele:  $L_i$  (poziția de început a neonului  $i$ ),  $R_i$  (poziția de sfârșit a neonului  $i$ ),  $P_i$  (numărul de antiradiații trimise de neonul  $i$ )
- pe următoarele  $p$  linii se afla șirul  $X_i$  (numărul de antiradiații necesare fiecărui tip de pepene pentru a deveni comestibil)

**Date de ieșire**

Fișierul de ieșire *pepeni.out* va conține  $p$  linii: pe linia  $i$  se va afișa "DA" în cazul în care pepenele de tipul  $i$  poate deveni comestibil și "NU" în caz contrar.

**Restricții și precizări:**

- $1 \leq q \leq 200$
- $1 \leq L[i], R[i], X[i] \leq n \leq 10.000$
- $1 \leq P[i] \leq 12.000$
- $1 \leq p \leq 100.000$

## Secțiunea 9-10 avansați

## Exemple

| pepeni.in   | pepeni.out                 | Explicație   |
|---|----------------------------|--|
| 4 2 2<br>1 3 1<br>3 4 2<br>2<br>3                         | DA<br>DA                   | Ambele tipuri de pepeni pot deveni comestibili astfel: pentru primul tip de va acționa neonul 2, iar pe pozițiile 3-4 se vor forma pepeni comestibili (căroră le-au fost aplicate 2 antiradiații); iar pentru al doilea tip trebuie acționate primele două neone, iar pe poziția 3 se va forma un pepene comestibil. |
| 8 3 5<br>1 3 4<br>2 4 3<br>4 7 1<br>1<br>3<br>5<br>7<br>2 | DA<br>DA<br>NU<br>DA<br>NU | Primul tip de pepeni devine comestibil acționând neonul 3;<br>Al doilea tip acționând neonul 2;<br>Al treilea tip nu poate deveni comestibil;<br>Al patrulea tip acționând neonele 1 și 2;<br>Al cincilea tip nu poate deveni comestibil.  |

**Timp maxim de execuție:** 0.2 secunde/test.

**Memorie totală disponibilă:** 1 MB, din care 1 MB pentru stivă

**Dimensiunea maximă a sursei:** 5 Kb

**PROBLEMA 4      PREMIU****100 puncte**

După 2 săptămâni de stat acasă, Mario observă că a rămas în urmă cu temele de la școală. Acesta decide să învețe mai întâi la matematică și română, unde are de lucrat **n**, respectiv **m** exerciții. El poate rezolva exercițiile unul câte unul, în ce ordine dorește. În plus, ca să fie mai motivat, profesorii lui Mario au creat “Premiul cel Mare”, pe care îl va primi doar dacă rezolvă minim **a** exerciții de matematică consecutive sau minim **b** exerciții de română consecutive.

**Cerință**

Găsiți numărul de moduri în care Mario poate să rezolve exercițiile, astfel încât să câștige “Premiul cel Mare”.

**Date de intrare**

În fișierul *premiu.in* se găsesc pe prima linie numerele **n**, **m**, **a**, **b**, separate prin câte un spațiu, cu semnificația din enunț.

**Date de ieșire**

În fișierul *premiu.out* se va afișa pe prima linie numărul de moduri în care Mario poate să rezolve exercițiile, astfel încât să câștige “Premiul cel mare”.

**Rezultatul va fi afișat modulo 666013.**

## Secțiunea 9-10 avansați

## Restricții și precizări

- $1 \leq n, m \leq 1000$
- $2 \leq a, b \leq 10$

## Exemple

| premiu.in | premiu.out | Explicații   |
|-----------|------------|--|
| 1 2 2 2   | 2          | Cele 2 moduri posibile sunt: <i>MRR</i> , <i>RMM</i> .<br>Observăm ca <i>RMR</i> nu respecta condițiile. |
| 5 5 2 2   | 250        |  |

**Timp maxim de execuție:** 0.1 secunde/test

**Memorie totală disponibilă:** 16 MB, din care 16 MB pentru stivă

**Dimensiunea maximă a sursei:** 5 KB