

## Secțiunea 9-10 începători

## PROBLEMA 1      ROBAB

100 puncte

O planșă de formă dreptunghiulară este împărțită în  $n$  rânduri egale și pe fiecare rând în  $m$  casete egale. În fiecare casetă este scris un număr natural nenul cu maxim 5 cifre.

Pe marginea planșei, în dreptul primului rând, de o parte și de alta, sunt așezați doi roboți, A și B. A se deplasează începând de la prima casetă, spre dreapta, iar B se deplasează de la ultima casetă, spre stânga, respectând următoarele reguli:

- o deplasare presupune ca un robot să culegă toate numerele aflate pe rândul curent, până la întâlnirea primului număr prim, în direcția în care se deplasează robotul respectiv;
- dacă pe un rând numărul de numere prime este impar atunci A va efectua deplasarea, iar B nu va face nimic;
- dacă pe un rând numărul de numere prime este par nenul atunci B va efectua deplasarea, iar A nu va face nimic;
- dacă pe un rând nu se găsesc numere prime atunci nici un robot nu face nimic;
- după testarea unui rând ambii roboți trec pe pozițiile inițiale de la capetele rândului următor;
- se vor testa astfel toate rândurile planșei.

**Cerință**

Cunoscând dimensiunea planșei și valorile scrise în casete să se determine pentru fiecare robot câte deplasări efectuate și suma numerelor culese.

**Date de intrare**

Fișierul **robab.in** are următoarea structură:

- pe prima linie două numere naturale  $n$  și  $m$ , reprezentând dimensiunile planșei;
- pe următoarele  $n$  linii câte  $m$  numere naturale nenule, separate prin spațiu, reprezentând valorile scrise în casete.

**Date de ieșire**

Fișierul **robab.out** are următoarea structură:

- pe prima linie numărul de deplasări efectuate de robotul A și suma numerelor culese de acesta, separate printr-un spațiu;
- pe a doua linie numărul de deplasări efectuate de robotul B și suma numerelor culese de acesta, separate printr-un spațiu.

**Restricții și precizări:**

- $2 \leq n, m \leq 400$ ;
- se consideră că un robot a efectuat **o deplasare** atunci când a plecat din poziția sa inițială și a cules numere pe rândul curent.

**Exemplu:**

robab.in	robab.out	Explicație
3 4 1 2 3 4 6 9 7 10 12 8 26 28	1 15 1 4	- pe primul rând s-a deplasat B și a cules 4 - pe al doilea rând s-a deplasat A și a cules 15 - pe al treilea rând nu s-a deplasat nici un robot

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

**Memorie totală disponibilă:** 2 MB din care 2 MB pentru stivă

**Dimensiunea maximă a sursei:** 5 Kb

## Secțiunea 9-10 începători

## PROBLEMA 2 ALUNE

100 puncte

În pădurea magică plină de alune trăia o veveriță mică pe care o chema VeVe. De când era o veveriță și mai mica, tot ce și-a dorit a fost să devină grasă, dorind mereu să mănânce mai multe alune. Totuși, fiind foarte puturoasă, VeVe nu mânca alune decât dintr-un singur copac.

Pădurea magică este alcătuită din  $n * m$  copaci, distribuiți pe  $n$  linii și  $m$  coloane, fiecare cu un număr nenul de alune. Se știe faptul că în fiecare zi, în mod magic, apar schimbări asupra copacilor: tuturor copacilor dintre pozițiile din dreptunghiul cu colțul stângă sus  $i1, j1$  și  $i2, j2$  colțul dreapta jos li se modifică numărul de alune prin adaugarea unei valori  $x$ .

**Cerință**

Dându-se  $n$  și  $m$ , numărul de alune inițial din fiecare copac și schimbările care au loc, ajutați-o pe VeVe să determine care este numărul maxim de alune dintr-un copac.

**Date de intrare**

Fișierul de intrare **alune.in** conține pe primul rând două numere:  $n$  și  $m$  cu semnificația de mai sus. Pe următoarele  $n$  linii se află câte  $m$  numere, fiecare număr reprezentând numărul de alune inițial din fiecare copac. Pe următoarea linie se află un singur număr  $z$  reprezentând numărul de schimbări. Următoarele  $z$  linii descriu schimbările. Pe fiecare linie se află câte 5 numere  $i1 j1 i2 j2 x$  care descriu schimbarea: copacilor din dreptunghiul determinat de colțul stânga sus  $i1, j1$  și colțul dreapta jos  $i2, j2$  li se modifică numărul de alune cu  $x$ .

**Date de ieșire**

Fișierul de ieșire **alune.out** conține un singur număr, reprezentând valoare maxima a numărului de alune dintr-un copac.

**Restricții și precizări**

- $1 \leq n, m \leq 1000$
- $z \leq 10000$
- $x$  este o valoare de tip întreg
- În orice moment numărul de alune ale unui copac poate avea cel mult 18 cifre (pozitiv sau negativ) VeVe vă mulțumește că ați ajutat-o să se îngrașe.

**Exemplu**

alune.in	alune.out	Explicație
3 4 3 8 9 10 10 9 1 4 6 1 4 5 2 1 1 3 4 10 3 3 3 4 5	20	Matricea după prima schimbare: <b>13 18 19 20</b> <b>20 19 11 14</b> <b>16 11 14 15</b> Matricea după cea de a doua schimbare: 13 18 19 20 20 19 11 14 16 11 <b>19 20</b> La finalul schimbărilor maximul este 20.

**Timp maxim de execuție:** 0.4 secunde/test

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

**Dimensiunea maximă a sursei:** 5 Kb

## Secțiunea 9-10 începători

## PROBLEMA 3 DIVIMP

100 puncte

Lui Budi îi place foarte mult să lucreze la matematica. De aceea, el a primit că temă N probleme, numerotate de la 1 la N, probleme ce trebuie rezolvate cât mai rapid. Pentru că în tema primită se mai găsesc și probleme ușoare, Budi are de gând să rezolve numai acele probleme cu un grad ridicat de dificultate, numite “probleme magice”. O problema este “magică”, doar dacă este îndeplinită una dintre următoarele 2 condiții:

1. indicele corespunzător problemei are un număr par de divizori;
2. indicele corespunzător problemei este divizibil cu 9 și are un număr impar de divizori;

Din păcate, chiar dacă Budi termina tema, acesta nu se poate relaxa imediat. De fiecare dată, prietenii lui apropiați îi pun diferite întrebări de genul: “Salut! Îmi spui și mie te rog ce probleme ai făcut de la problema A la problema B?”.

**Cerință**

Ajutați-l pe Budi să răspundă rapid la aceste mesaje, pentru a se putea odihni!

**Date de intrare**

Fișierul de intrare **divimp.in** va conține pe prima linie numărul N, reprezentând numărul de probleme pe care Budi le-a primit ca tema. Pe linia a doua se va afla un număr T, reprezentând numărul de prieteni ai lui Budi, urmat de T linii, fiecare linie conținând două numere: A și B. ( $A \leq B$ ).

**Date de ieșire**

Fișierul de ieșire **divimp.out** va conține T linii, fiecare dintre ele reprezentând răspunsul la întrebarea: câte probleme a rezolvat Budi de la problema A la problema B.

**Restricții și precizări**

$1 \leq N \leq 1.000.000$ ;

$1 \leq T \leq 1.000.000$ ;

$1 \leq A \leq B \leq N$ ;

**Exemplu**

divimp.in	divimp.out	Explicații
100	7	Budi are de rezolvat 100 de probleme.
5	54	De la problema 84 pana la problema 90 acesta va rezolva problemele <b>84 85 86 87 88 89 90</b> , adica toate problemele.
84 90	56	
44 100	14	De la problema 44 pana la problema 100, acesta va rezolva 54 de probleme.
9 68	20	
86 100		De la problema 9 la 68, Budi rezolva 56 de probleme.
35 55		De la problema 86 la problema 100, Budi va rezolva problemele <b>86 87 88 89 90 91 92 93 94 95 96 97 98 99</b> , adica 14 probleme.
		In final, de la problema 35 pana la 55 Budi rezolva doar 20 de probleme: <b>35 36 37 38 39 40 41 42 43 44 45 46 47 48 50 51 52 53 54 55</b> .

**Timp maxim de execuție:** 0.3 secunde/test

**Memorie totală disponibilă:** 5 MB din care 2 MB pentru stivă

**Dimensiunea maximă a sursei:** 5 Kb

## Secțiunea 9-10 începători

## PROBLEMA 4 YOUME

100 puncte

În clasa lui Octavian sunt  $N$  elevi, numerotați de la 1 la  $N$ . În clasa vecină, există  $M$  elevi, notați cu indici numere naturale  $v[1], v[2], \dots, v[M]$ . Spunem că elevul cu indicele „ $i$ ” este prieten cu elevul cu indicele „ $j$ ” dacă  $\text{cmmdc}(i, j) = i$ , unde  $\text{cmmdc}$  reprezintă cel mai mare divizor comun.

**Cerință**

Se dau  $Q$  elevi din clasa lui Octavian. Să se afișeze pentru fiecare numărul său de prieteni din cele 2 clase.

**Date de intrare**

Din fișierul **youme.in**, se citesc pe prima linie numerele naturale  $N$ , cu semnificația din enunț. Pe următoarea linie se află  $M$ , urmat de  $M$  numere naturale, reprezentând indicii elevilor din clasa vecină clasei lui Octavian. Pe a 3-a linie se află numărul natural  $Q$ , cu semnificația din enunț. Pe următoarea linie, se află  $Q$  numere ( $Q[1], Q[2], \dots, Q[q]$ ), reprezentând indicii elevilor din clasa lui Octavian, cărora trebuie să le aflați numărul de prieteni.

**Date de ieșire:**

În fișierul **youme.out** se vor afișa  $Q$  valori, pe o singură linie separate prin spațiu. A  $K$ -a valoare va reprezenta numărul de prieteni al elevului cu indicele  $Q[k]$ .

**Restricții și precizări:**

- $1 \leq N \leq 1000000$
- $1 \leq M \leq 1000$
- $1 \leq Q \leq 1000$ .
- $1 \leq V[k] \leq 2^{100}$ , unde  $2^{100}$  reprezintă 2 la puterea 100, pentru orice  $k$  între 1 și  $M$ .
- **Indicii elevilor din clasa lui Octavian pot coincide cu cei din clasa vecină.**
- **Dacă elevul „ $i$ ” este prieten cu elevul „ $j$ ”, nu înseamnă că elevul „ $j$ ” este prieten cu elevul „ $i$ ”**
- $1 \leq N \leq 1000$ ,  $M=0$  și  $1 \leq Q \leq 1000$ , pentru teste în valoare de 20 de puncte
- $1 \leq V[k] \leq 1000000$ , pentru alte 15 puncte.

**Exemple**

youme.in	youme.out	Explicatii
10 1 2 3 1 2 3	10 5 2	Elevul cu indicele 1 este prieten cu elevii 2,3,4,5,6,7,8,9,10 din clasa lui Octavian și cu elevul 2 din clasa vecină. Elevul cu indicele 2 este prieten cu elevii 4,6,8,10 din clasa lui Octavian și cu elevul 2 din clasa vecină. Elevul cu indicele 3 este prieten cu elevii 6,9 din clasa lui Octavian.
20 5 100 200 300 400 500 2 10 5	6 8	

**Timp maxim de execuție:** 0.4 secunde/test.

**Memorie totală disponibilă:** 8 MB, din care 2 MB pentru stivă

**Dimensiunea maximă a sursei:** 5 Kb