

PROBLEMA 1

100 puncte

ARTIFICII

Cu ocazia Zilelor Orașului se organizează un frumos foc de artificii. Organizatorii au pus la dispoziție artificii în cutii sigilate, în care artificiile sunt numerotate cu numere consecutive de la C1 la C2. Pentru ca spectacolul oferit să fie cât mai animat, ei au cerut producătorului ca în fiecare cutie artificiile să aibă puteri diferite. Patronul fabricii de artificii, domnul Mircea, a găsit o soluție pentru solicitarea organizatorilor și a asociat fiecărui artificiu o putere legată de numărul lui de ordine. Astfel, puterea unui artificiu este egală cu numărul de numere prime distincte pe care le generează numărul lui de ordine. De exemplu, artificiuul cu numărul 15 are puterea 2 deoarece el generează pe 3 și pe 5.

Cerință

Determinați câte artificii de putere P se află în fiecare cutie trimisă de producător.

Date de intrare

Fișierul **artificii.in** conține cel mult 100000 de linii cu câte trei numere naturale C1, C2 și P, cu semnificația de mai sus.

Date de ieșire

Fișierul **artificii.out** conține, câte unul pe un rând, numerele cerute.

Restricții și precizări:

- $1 \leq C1 < C2 \leq 1000000$;
- $0 \leq P \leq 2000$.

Exemplu:

artificii.in	artificii.out	Explicație
1 15 1	9	artificiile 2,3,4,5,7,8, 9,11,13 au puterea 1
1 15 2	5	artificiile 6,10,12,14,15 au puterea 2
1 15 3	0	nu sunt artificii cu puterea 3
1 15 0	1	artificiul 1 are puterea 0

Timp maxim de execuție: 0,6 secunde/test.

Memorie totală disponibilă: 32 MB, din care 20 MB pentru stivă

Dimensiunea maximă a sursei: 5 KB

PROBLEMA 2

100 puncte

BRUTAR

Renumitul nostru brutar a avut azi noapte un vis tare ciudat: acesta trăia într-un univers paralel în care nu omul îl mănâncă pe blat ci blatul îl mănâncă pe om... (eh, poate nu chiar atât de paralel). Astfel, brutarul nostru a fost atacat de blatul pe care tocmai îl pregătise (pentru prăjituri, evident) și a încercat să scape. Acesta a ieșit din brutărie și a ajuns în fața unui câmp de formă dreptunghiulară, cu dimensiunile cunoscute, ce poate fi împărțit în celule elementare cu latura de o unitate (exact ca o matrice!). Acesta poate intra pe câmp prin orice celulă a primei linii și trebuie să

Ploiești, 28 mai 2016

ajungă în orice celulă a ultimei linii (blatul se va întări până va ajunge acolo). Unele celule îi sunt inaccesibile din cauza diverselor obstacole (pietre, pomi, gropi...etc)

Brutarul nostru se poate deplasa în 6 moduri:

- Din căsuța curentă în cele adiacente (Nord, Vest, Sud, Est)
- Două mișcări speciale ce pot varia.

Mutările speciale vor fi citite din fișier și o mutare se va codifica astfel: $x'A' y'B'$, unde x și y sunt numere naturale nenule iar 'A' și 'B' sunt două caractere ce codifică direcția ('A' poate fi 'N' sau 'S' de la Nord respectiv 'Sud' iar 'B' poate fi 'E' sau 'V' de la Est respectiv Vest)

Ex. 5N 2V codifică mutarea 5 poziții spre Nord și 2 poziții spre Vest. (din poziția (x,y) ajunge în poziția $(x-5, y-2)$)

O mutare specială se poate face dacă celula destinație nu este ocupată de un obstacol și dacă nu implică ieșirea brutarului din matrice.

Cerința

Brutarul vă roagă să îi specificați un traseu cu număr minim de celule parcurse, ce pornește de pe prima linie și se termină pe ultima linie, pentru a nu fi blătuit (mâncat de blat).

Date de intrare

Pe prima linie a fișierului *brutar.in* se vor afla două numere naturale separate prin spațiu: N și M reprezentând numărul de linii și numărul de coloane ale câmpului. Urmează N linii a câte M caractere 'X' și 'O' unde 'X' reprezintă obstacol iar 'O' reprezintă zonă liberă.

Ultimele două linii reprezintă codificarea celor două mutări speciale, câte o mutare pe o linie.

Date de ieșire

Fișierul *brutar.out* va conține, pe prima linie, numărul T de celule al unui traseu de lungime minimă. Următoarele T linii vor conține descrierea traseului, câte o celulă pe o linie, prin coordonatele ei, în ordinea în care au fost parcurse.

Exemplu

brutar.in	brutar.out	Explicație
14 10	9	Drumul este colorat în galben
XXXXXXXXXXX	1 7	
OXOOOXOOO	3 6	XXXXXXXXOXXX
OXXOXXOXO	5 7	OXOOOXOOO
OXOOOXXXX	7 6	OXXOXXOXO
OOOXXOXOX	9 5	OXOOOXXXX
OXOXOXXOO	10 5	OOOXXOXXO
XOXOOOXXO	12 4	OXOXOXXOO
OXXOXOOOX	12 3	XOXOXXOXO
XOOXOXOXO	14 2	OXXOXOOOX
OXOXOXOXO		XOOXOXXOXO
OOOXXOXXO		OXOXOXXOXO
XOOXOXOXO		OOOXXOXXO
OXOOOXXXX		XOOXOXXOXO
2S 1V		OXOOOXXXX
2S 1E		XOXXOXOXO

Restricții și precizări:

- $1 \leq N, M \leq 1.000$
- $1 \leq x \leq N$
- $1 \leq y \leq M$
- Se garantează că mereu va exista cel puțin un drum care pornește de pe prima linie și ajunge pe ultima linie a matricei.
- Dacă există mai multe soluții cu număr minim de pași, se poate afișa oricare dintre acestea.

Timp maxim de execuție: 0.5 secunde/test**Memorie totală disponibilă:** 16MB și 4MB pentru stivă**Dimensiunea maximă a sursei:** 5Kb**PROBLEMA 3****100 puncte****PRIM**

Un număr natural nenul se numește “p-prim” dacă el se descompune în p moduri ca produs de doi factori primi între ei. De exemplu, numărul 60 este 4-prim deoarece $60 = 1 \cdot 60 = 4 \cdot 15 = 5 \cdot 12 = 20 \cdot 3$, iar numărul 7 este 1-prim.

Pentru un interval închis $[a, b]$ să se determine câte numere p-prime aparțin intervalului. De exemplu intervalul $[7, 20]$ conține numerele 2-prime: 10, 12, 14, 18, 20.

Date de intrare

Din fișierul de intrare prim.in se citesc de pe prima linie două numere naturale N și P și de pe următoarele N linii câte două numere ce reprezintă capetele unui interval.

Date de ieșire

În fișierul de ieșire prim.out se va scrie pe prima linie intervalul cu cele mai multe numere p-prime. Extremitățile intervalului se vor afișa în ordine crescătoare. Dacă există mai multe intervale cu același număr se va afișa ultimul interval citit. Dacă nu există niciun interval se va afișa mesajul nu exista

Restricții și precizări $3 \leq N \leq 1000$ $1 \leq P \leq 1000$ $1 \leq a[i], b[i] \leq 33000$, unde $a[i]$ și $b[i]$ sunt capetele intervalelor, $i=1, 2, \dots, N$ **Exemplu**

prim.in	prim.out	Explicație
4 2 20 7 5 10 35 39 3 4	7 20	Intervalul $[7, 20]$ conține numerele 2-prime: 10, 12, 14, 15, 18, 20 ; intervalul $[5, 10]$ conține numerele 2-prim 6 și 10; intervalul $[35, 39]$ conține numere 2-prime 35, 36, 38, 39; intervalul $[3, 4]$ nu conține niciun număr 2-prim.

Timp maxim de execuție: 0,1 secundă/test.

Memorie totală disponibilă: 4 MB, din care 2 MB pentru stivă

Dimensiunea maximă a sursei: 5 KB.

PROBLEMA 4

100 puncte

PROIECȚIE

Colegul vostru Macarie este fascinat de geometria în spațiu și de informatică. La unul din concursurile la care a participat a avut de rezolvat următoarea problemă:

Cunoscând coordonatele a N puncte în plan, generați toate segmentele care se pot forma prin unirea a două puncte și determinați valoarea maximă a sumei lungimilor proiecțiilor fiecărui segment pe cele 2 axe de coordonate.

Date de intrare

Pe prima linie a fișierului proiectie.in se află un număr N , reprezentând numărul de puncte. Pe următoarele N linii sunt câte 2 numere întregi separate prin câte un spațiu, reprezentând în ordine, coordonatele X și Y ale celor N puncte.

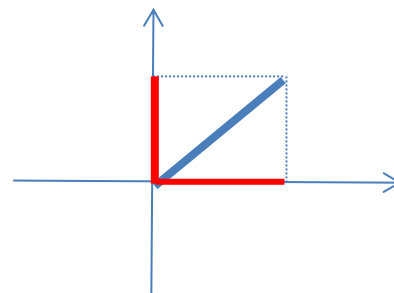
Date de ieșire

Fișierul proiectie.out va conține o linie cu un singur număr întreg.

Restricții și precizări

$-1\ 000\ 000 \leq X, Y \leq 1\ 000\ 000$

$2 \leq N \leq 100\ 000$



Exemplu

proiectie.in	proiectie.out	Explicații
4	2	Se pot forma $\text{Comb}(4,2)$ segmente =6.
0 0		Dintre acestea suma maxima cerută se obține pentru
0 1		segmentul cu capetele în punctele de coordonate (0,0) și (1,1).
1 0		
1 1		

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.