

CLASA a IX-a  
NECTARPROBLEMA 1  
100 p

Albinuța locuiește într-un stup și are un fagure rezervat. Fagurele ei conține 1000 de celule pe verticală, numerotate de sus în jos, grupate în  $C$  coloane pe orizontală, numerotate de la stânga la dreapta. Ea știe foarte bine ce tip de nectar a depus în fiecare celulă, iar acum își dă seama că trebuie să depună cam mult efort să adune un anumit tip de nectar pentru producerea mierii. În cazul în care Albinuța coboară la nivelul  $k$ , ea poate aduna tot nectarul de care are nevoie din celulele de la nivelul 1 până la nivelul  $k$ , de pe coloana pe care a coborât și de pe coloanele vecine (din stânga și din dreapta). Ea nu poate coborî decât direct pe verticală și acest lucru o obosește foarte tare. Cunoscând celulele în care a depus nectarul pentru tipul de miere pe care vrea să-l fabrice, Albinuța vă roagă s-o ajutați să adune tot nectarul care-i trebuie, dar suma nivelelor pe care trebuie să le coboare să fie minimă.

**Cerință:**

Scrieți un program care să determine suma minimă a nivelelor până la care trebuie să coboare Albinuța pentru a aduna tot nectarul necesar.

**Date de intrare**

Prima linie a fișierului de intrare `nectar.in` conține două numere naturale  $C$  și  $N$ , separate printr-un spațiu, reprezentând numărul coloanelor de celule și respectiv numărul celulelor care conțin nectarul pe care trebuie să-l adune Albinuța.

Următoarele  $N$  linii conțin câte două numere naturale  $a$   $b$ , separate printr-un spațiu, reprezentând numărul coloanei, respectiv nivelul de unde trebuie luat nectarul.

**Date de ieșire**

Fișierul de ieșire `nectar.out` va conține un singur număr natural reprezentând suma minimă a nivelelor până la care trebuie să coboare Albinuța pentru a aduna tot nectarul necesar.

**Restricții și precizări**

$$1 \leq C \leq 10.000$$

$$1 \leq N \leq 50.000$$

**Exemple**

<code>nectar.in</code>	<code>nectar.out</code>	<i>Explicație</i>
10 4 5 4 1 1 6 2 3 8	11	Albinuța coboară astfel: - pe coloana 1 la nivelul 1 - pe coloana 4 la nivelul 8 - pe coloana 6 la nivelul 2

**Timp maxim de execuție:** 0.025 secunde

**Total memorie disponibilă** 5MB.

**CLASA a IX-a**  
**CHIPDALE****PROBLEMA 2**  
**100 p**

Chip și Dale, celebrele veverițe din filmele lui Disney, au adunat toată vara o mulțime de ghinde și alune, pe care le-au depozitat în cele  $C$  cutii pe care le-au avut la dispoziție. Fiecare cutie are două sertare: unul pentru ghinde și celălalt pentru alune. Veverițele, istețe cum sunt, au etichetat cutiile cu numere distincte de la 1 la  $C$  și știu câte ghinde și câte alune conține fiecare cutie, numai că în depozitul de iarnă ele au calculat că nu încap decât jumătate din cutiile cu hrană și încă o cutie. În zonă meteorologii anunță o iarnă foarte grea, iar veverițele nu se vor putea deplasa spre depozitul de vară și vă roagă pe voi să alegeți acele cutii care conțin nu mai puțin decât jumătate din cantitatea totală de ghindă și nu mai puțin decât jumătate din cantitatea totală de alune. Certărețe cum sunt, ele vor să repartizeze în mod egal toate alunele și ghindele din cutiile selectate în cămările din depozitul de iarnă al scorbirii. Știind că ele își pot construi repede oricâte cămări, încercați să le împăcați determinând în câte moduri pot împărți întreaga cantitate de provizii din cutiile selectate, în mod egal în fiecare cămară.

**Cerință**

Scrieți un program care să aleagă  $N$  cutii din cele  $C=2*N-1$  cutii, pentru a îndeplini condiția cerută de veverițe, precum și numărul de posibilități de repartizare a proviziilor în cămări.

**Date de intrare**

Prima linie a fișierului `chipdale.in`, conține numărul natural  $C$ , reprezentând numărul cutiilor cu provizii adunate peste vară, următoarele  $C$  linii conțin câte două numere naturale separate prin spațiu  $a_i$   $g_i$  reprezentând numărul de alune, respectiv de ghinde aflate în cutia cu eticheta  $i$ .

**Date de ieșire**

În fișierul `chipdale.out`, prima linie va conține cuvântul “DA”, dacă este posibil să se aleagă  $N$  cutii sau “NU”, în caz contrar. În cazul în care răspunsul este afirmativ, pe a doua linie se vor scrie  $N$  numere naturale separate printr-un spațiu, reprezentând etichetele cutiilor alese, iar pe a treia linie două numere naturale  $h$   $p$ , separate prin spațiu, reprezentând numărul total de provizii din cutiile selectate, respectiv numărul posibilităților de repartizare a acestora în mod egal în fiecare cămară.

**Restricții și precizări**

$$1 \leq a_i, g_i \leq 10^9$$

$$1 \leq C \leq 20000$$

**Exemplu**

<code>chipdale.in</code>	<code>chipdale.out</code>
3	DA
10 15	1 3
15 12	63 6
20 18	

**Timp maxim de execuție:** 0.1 secunde

**Total memorie disponibilă** 5 MB.

**CLASA a X-a**  
**NAO****PROBLEMA 1**  
**100 p**

Pe planeta LISIOM se află o suprafață destul de mare unde au fost depozitate materiale periculoase, aflate acum în descompunere, care produc perturbări ale mediului planetei, conducând la schimbări climatice. Cercetătorii de pe planetă au creat un robot NAO, pe care l-au înzestrat cu un dispozitiv pentru recunoașterea tuturor materialelor periculoase și neutralizarea efectelor descompunerii lor.

Suprafața unde au fost depozitate materialele are  $4 \cdot n$  metri lungime pe direcția nord-sud și  $4 \cdot m$  metri lățime pe direcția vest-est. Pentru ca materialele să nu se influențeze reciproc, suprafața a fost împărțită în  $n \cdot m$  pătrate identice, fiecare material ocupând exact un pătrat, iar pătratele au fost separate de  $n+1$  linii orizontale  $l_i$ , ( $0 \leq l_i \leq n$ ) și  $m+1$  linii verticale  $l_j$  ( $0 \leq l_j \leq m$ ). Pentru a neutraliza efectele descompunerii materialelor, NAO poate fi plasat în oricare din intersecțiile liniilor și nu se poate re poziționa până când nu neutralizează toate materialele. Pentru neutralizarea unui material aflat în pătratul  $i, j$  ( $1 \leq i \leq n$  și  $1 \leq j \leq m$ ), NAO are nevoie de un timp direct proporțional cu pătratul distanței euclidiene din punctul în care se află până în centrul pătratului unde se află materialul și cu un coeficient  $c[i][j]$  ce caracterizează rezistența la neutralizare a materialului respectiv.

Cercetătorii vă roagă să-i ajutați să aleagă un punct unde să-l plaseze pe NAO, astfel încât efectele descompunerii materialelor să fie neutralizate în cel mai scurt timp posibil.

**Cerință**

Scrieți un program care să determine timpul minim de neutralizare și coordonatele ( $l_i$ ,  $l_j$ ) unde trebuie plasat NAO.

**Date de intrare**

Pe prima linie a fișierului `nao.in` se află două numere întregi,  $n$  și  $m$  reprezentând dimensiunile suprafeței de depozitare, iar următoarele  $n$  linii conțin câte  $m$  numere întregi,  $c[i][j]$  – coeficienții de rezistență la neutralizare al materialului aflat în pătratul  $i, j$ .

**Date de ieșire**

Pe prima linie a fișierului `nao.out` se va scrie un număr întreg reprezentând timpul minim în care toate materialele sunt neutralizate, iar pe a doua linie, intersecția  $l_i$ ,  $l_j$  ( $0 \leq l_i \leq n$ ,  $0 \leq l_j \leq m$ ) unde poate fi plasat NAO ca să obțină acest timp de neutralizare. Dacă există mai multe poziții optime, scrieți punctul cu  $l_i$  mai mic, iar dacă există mai multe astfel de puncte, scrieți punctul cu  $l_j$  mai mic.

**Restricții și precizări**

$$1 \leq n, m \leq 1000$$

$$0 \leq c[i][j] \leq 100000$$

$$1 \leq i \leq n, 1 \leq j \leq m$$

**Exemple:**

<code>nao.in</code>	<code>nao.out</code>	Explicație
2 3 3 4 5 3 9 1	392 1 1	Timpul total de neutralizare este: $3 \cdot 8 + 3 \cdot 8 + 4 \cdot 8 + 9 \cdot 8 + 5 \cdot 40 + 1 \cdot 40 = 392$
3 4 1 0 0 0 0 0 3 0 0 0 5 5	240 2 3	

**Timp maxim de execuție/test:** 0.05 secunde

**Total memorie disponibilă** 4 MB

**CLASA a X-a**  
**BREE****PROBLEMA 2**  
**100 p**

Roboțelul BREE lucrează la fabrica de bere și are în program două operații: încarcă o sticlă de bere și pune capac la o sticlă plină cu bere. El primește comenzile printr-un șir de numere naturale nenule pe care le memorează în format binar. Dacă bitul cel mai puțin semnificativ al numărului primit este nul, el încarcă o sticlă cu bere, iar dacă este nenul, atunci pune capac unei sticle. Programul generatorului de comenzi are niște erori și câteodată BREE nu găsește sticle pline cărora să le pună capac sau lasă sticle pline fără capac. Pentru a evita aceste erori, din șirul primit trebuie păstrate doar cele mai lungi secvențe corecte de comenzi.

**Cerință**

Scrieți un program care, citind un șir de numere naturale, determină lungimea maximă a unei secvențe corecte de operații pentru BREE, câte astfel de secvențe se află în șir și pozițiile de început ale secvențelor corecte maximale.

**Date de intrare**

În fișierul **bree.in** este scris pe prima linie un șir de numere naturale nenule  $x_i$ , care se termină cu 0.

**Date de ieșire**

Pe prima linie a fișierului de ieșire **bree.out** se scriu două numere întregi **L** și **N**, reprezentând lungimea maximă a unei secvențe corecte de operații și numărul secvențelor cu această proprietate. Dacă nu există nici o secvență corectă se va scrie: **1 0**, dacă se încarcă cel puțin o sticlă sau **0 0**, dacă nu se încarcă nici o sticlă.

Pe a doua linie a fișierului se află **N** numere naturale reprezentând pozițiile din șir unde încep secvențele maximale corecte.

**Restricții și precizări:**

Numărul numerelor din fișierul de intrare nu depășește  $10^6$ .

$$1 \leq x_i \leq 10^9$$

**Exemplu:**

<b>bree.in</b>	<b>bree.out</b>	<b>Explicație</b>
5 4 6 8 3 7 9 13 12 44 3 6 11 9 0	6 2 2 9	Prin comenzile din poziția 2 până în poziția 7, tuturor sticlelor încărcate li se pune capac, la fel pentru secvența de comenzi din poziția 9 până în poziția 14
51 73 86 22 0	1 0	Încarcă două sticle fără să le pună capace
45 123 57 81 100001 0	0 0	Nu încarcă nici o sticlă

**Timp maxim de execuție:** 0.1 secunde

**Total memorie disponibilă** 10 MB.

**CLASA a XI-a**  
**POD****PROBLEMA 1**  
**100 p**

Ca Împăratul Verde să i-o dea pe fata cea mică de soție, Făt Frumos trebuie să placheze podul care leagă Castelul Împăratului de Pădurea Minune, în care trăiește Cerbul cu steaua în frunte. Din când în când Cerbul trebuie să vină la castel după hrană și își pune viața în pericol traversând autostrada pe care circulă zmeii cu mașini de lux, pentru că de-a lungul timpului podul s-a rupt și nu mai poate fi utilizat.

Suprafața podului este de  $n \times m$  metri pătrați. Pentru placare Făt Frumos poate folosi trei tipuri de plăci: tipul **a** de dimensiuni  $1m \times 2$ , tipul **b** de dimensiuni  $2m \times 1m$ , și tipul **c** de dimensiuni  $2m \times 2m$ . Plăcile sunt aduse de cei șapte pitici, care lucrează din greu la exploatarea bazaltului de dincolo de pădure.

Ajutați-l pe Făt Frumos să afle dacă poate placa podul cu un astfel de set de plăci, și dacă este posibil, găsiți una din modalitățile de a face acest lucru. Făt Frumos poate să nu folosească toate plăcile avute la dispoziție.

**Cerință**

Scrieți un program care să determine una din modalitățile de placare a podului.

**Date de intrare**

Pe prima linie a fișierului de intrare **pod.in** sunt cinci numere: **n**, **m**, **a**, **b**, **c** separate prin spațiu, cu semnificația **n** și **m** – dimensiunile podului, **a** – numărul plăcilor de dimensiune  $1m \times 2m$ , **b** – numărul plăcilor de dimensiune  $2m \times 1m$  și **c** – numărul plăcilor de dimensiune  $2m \times 2m$ . Atenție, plăcile nu se pot răsuci.

**Date de ieșire**

Dacă placarea nu se poate realiza cu plăcile existente, prima linie a fișierului **pod.out** va conține cuvântul „**NU**”. În caz contrar, prima linie va conține cuvântul „**DA**”, iar pe următoarele **n** linii, câte **m** litere mici latine. Două pătrate cu laturi comune trebuie să conțină aceeași literă dacă aparțin aceleiași plăci și litere diferite în caz contrar. Plăcile distincte pot fi marcate cu una și aceeași literă (vezi exemplul). Dacă răspunsul nu este unic, scrieți orice soluție de placare corectă.

**Restricții și precizări**

$$1 \leq n \leq 100; 1 \leq m \leq 100$$

$$0 \leq a \leq 10000; 0 \leq b \leq 10000; 0 \leq c \leq 10000$$

**Exemplu**

pod.in	pod.out
2 6 2 2 1	DA aabcca aabdda
1 1 100 100 100	NU

**Timp maxim de execuție:** 0.01 secunde

**Total memorie disponibilă** 3 MB

**CLASA a XI-a**  
**DIVY****PROBLEMA 2**  
**100 p**

Un grup de cercetători spațiali vor să exploreze planeta Divy din galaxia Nprim. Ei au aflat de la Dr. Beverly Crusher de pe nava Enterprise că pot colecta mai multe microorganisme pentru a fi analizate pe Terra, dacă folosesc niște containere speciale. Pentru transportul în siguranță, microorganismele colectate trebuie repartizate în număr egal în containere. Există exact  $n$  moduri diferite de a repartiza microorganismele în containere. Pentru prelucrarea informațiilor lt. DATA are nevoie de numărul minim de microorganisme ce trebuie repartizate în containere.

**Cerință**

Scrieți un program prin care să determinați rapid acest număr

**Date de intrare**

Pe prima linie a fișierului de intrare **divy.in** se află un număr natural  $n$ , reprezentând numărul de moduri în care pot fi repartizate microorganismele în containere.

**Date de ieșire**

Fișierul **divy.out** va conține cel mai mic număr de microorganisme ce pot fi distribuite în mod egal în containere.

**Restricții și precizări**

$1 \leq n \leq 1000$

**Exemplu**

<b>divy.in</b>	<b>divy.out</b>
6	12
5	16

**Timp maxim de execuție:** 0.02 secunde

**Total memorie disponibilă:** 3 MB.

**CLASA a XII-a**  
**BROTAC****PROBLEMA 1**  
**100 p**

Făt Frumos a reușit s-o ducă pe Ileana Cosânzeana în Castelul său, aflat în Pădurea Fermecată. Dar tot acolo are un castel și Zmeul Zmeilor, care demult încearcă să o răpească pe Ileana Cosânzeana. Pădurea poate fi reprezentată printr-o suprafață dreptunghiulară cu  $N \times M$  pătrate. Fiecare pătrat al suprafeței corespunde unei poziții de pe poteci sau unui obstacol (groapă, copac, baltă, mlaștină). Când Făt Frumos merge la vânătoare, Ilenei îi place să se plimbe prin pădure. Cunoscându-i obiceiul, Zmeul Zmeilor stă la pândă și încearcă s-o prindă pe picior greșit. Nici Zmeul Zmeilor și nici Ileana Cosânzeana nu se aventurează să treacă peste obstacole. Ei se pot deplasa dintr-un pătrat în unul din cele patru pătrate vecine dacă acestea corespund unei poziții de pe o potecă. Unele poziții de pe poteci se pot transforma de către Brotacul Vrăjitor, prieten cu Făt Frumos, în poziții obstacol.

**Cerință**

Ajutați Brotacul Vrăjitor să aleagă un număr minim de poziții ”transformabile”, prin schimbarea cărora Zmeul și Ileana să nu se poată întâlni.

**Date de intrare**

Prima linie a fișierului **brotac.in** conține două numere întregi  $N$  și  $M$  reprezentând numărul de pătrate de la nord la sud, respectiv de la vest la est ale suprafeței de pădure. Următoarele  $N$  linii conțin câte  $M$  caractere cu următoarea semnificație:

- P** pentru o poziție de pe o potecă
- C** pentru o poziție a unui obstacol
- \*** pentru o poziție P ”transformabilă” într-o poziție C de către Brotacul Vrăjitor.
- I** pentru poziția inițială a Ilenei
- Z** pentru poziția inițială a Zmeului

**Date de ieșire**

Pe prima linie a fișierului **brotac.out** se scrie numărul întreg  $R$ , reprezentând numărul minim de poziții care trebuie transformate. Pe următoarele  $R$  linii vor apărea câte 2 numere, reprezentând pozițiile alese pentru a fi transformate. Primul număr va fi între 1 și  $N$ , iar al doilea va fi între 1 și  $M$ .

**Restricții și precizări**

$1 \leq N, M \leq 100$

Rezultatul afișat  $R$  poate fi și 0.

Pe testele date va exista întotdeauna soluție.

Pozițiile inițiale ale Ilenei și Zmeului sunt implicit zone de potecă, ce nu pot fi transformate de Brotac.

**Exemplu**

<b>brotac.in</b>	<b>brotac.out</b>
4 4	1
IC CZ	3 3
PC*P	
P**P	
**CP	

**Timp maxim de execuție:** 0.4 secunde

**Total memorie disponibilă** 2 MB.

**CLASA a XII-a**  
**ZORIL****PROBLEMA 2**  
**100 p**

Un grup de cercetători ai vieții în spațiu vor să exploreze planeta Zoril din galaxia Nomogramm. Ei au aflat de la Dr. Beverly Crusher de pe nava Enterprise că pot colecta mai multe microorganisme pentru a fi studiate, dacă folosesc niște containere speciale. Pentru transportul în siguranță spre Terra, microorganismele colectate trebuie repartizate în număr egal în containere. Există exact  $n$  moduri diferite de a repartiza microorganismele în containere. Pentru prelucrarea informațiilor It. DATA are nevoie de numărul minim de microorganisme ce trebuie repartizate în containere.

**Cerință**

Scrieți un program prin care să determinați rapid acest număr

**Date de intrare**

Pe prima linie a fișierului de intrare `zoril.in` se află un număr natural  $n$ , reprezentând numărul de moduri în care pot fi repartizate microorganismele în containere.

**Date de ieșire**

Fișierul `zoril.out` va conține cel mai mic număr de microorganisme ce pot fi distribuite în mod egal în containere.

**Restricții și precizări**

$1 \leq n \leq 50000$

**Exemplu**

<code>zoril.in</code>	<code>zoril.out</code>
6	12
5	16

**Timp maxim de execuție:** 0.01 secunde

**Total memorie disponibilă** 3 MB