

Secțiunea 9-10 începători

PROBLEMA 1 CLATITE

100 puncte

Dorind să ia o pauză de la matematică, Budi s-a gândit să mănânce niste clătite, mai exact N clătite. Budi vrea să mănânce clătitele fie cu miere, fie cu gem de afine, fie cu sirop de arțar (acestea fiind singurele topping-uri pe care el le are în casă). Ajutați-l pe Budi să afle în câte moduri își poate umple clătitele, știind că:

- Budi poate pune miere pe o clătită cu indicele egal cu 1, sau după o clătită pe care s-a pus gem de afine sau sirop de arțar (adică nu vor exista două clătite cu miere pe poziții consecutive);
- Budi poate pune sirop de arțar pe o clătită cu indicele egal cu 1, sau după o clătită pe care s-a pus gem de afine.
- Budi poate pune gem de afine (acesta fiind mai puțin dulce) pe orice clătită, indiferent de topping-urile anterioare;
- O clătită va avea doar un singur tip de topping;
- Nu va exista o clătită fără niciun topping.

Cerință

Budi nu vrea să renunțe să se gândească la matematică nici în pauza de masă, așa că vă roagă să îl ajutați să afle numărul de moduri în care acesta poate umple clătitele.

Date de intrare

Fișierul *clatite.in* va conține pe prima linie un singur număr, anume numărul de clătite pe care Budi le-a gătit și vrea să le umple.

Date de ieșire

Fișierul *clatite.out* va conține pe o linie rezultatul cerut. Fiindcă e posibil ca rezultatul să fie un număr foarte mare, acesta va fi afișat modulo 20173333.

Restricții și precizări

- $2 \leq N \leq 2.500.000$;

Exemplu:

| clatite.in | clatite.out | Explicație |
|------------|-------------|---|
| 3 | 13 | Budi poate umple clătitele astfel: G-G-G , G-G-M, G-G-S, G-S-G, G-S-M, G-M-G, S-G-G, S-G-M, S-G-S, S-M-G, M-G-G, M-G-S, M-G-M G = gem de afine; M = miere; S = sirop de artar; |
| 156 | 8710781 | Da! Știm că sunt multe clătite. |

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 începători

PROBLEMA 2 Marte

100 puncte

Centrul Județean de Excelență a construit un robot autonom destinat explorării planetei Marte. Robotul trebuie să parcurgă trasee pe suprafața planetei și să preleveze anumite roci. Harta planetei este codificată printr-o matrice A cu N linii și coloane în care elementul $A[I][J]$ reprezintă înălțimea în metri a celulei de coordonate I și J. Dintr-o celulă de coordonate X și Y robotul se poate deplasa cu un pas în celulele învecinate pe linii sau coloane. El dispune inițial de o cantitate E de energie. Dacă la deplasarea cu un pas el întâlnește o celulă cu înălțime strict mai mare el va trebui să consume un număr de unități de energie egal cu câți metri va trebui să urce, iar dacă cantitatea rămasă de energie este insuficientă el nu se va putea deplasa în acea celulă. Dacă înălțimea celulei următoare este egală sau mai mică decât cea din care pleacă el nu va consuma din provizia de energie. În anumite celule din matrice se află roci pe care poate să le preleveze. Într-o celulă se pot afla mai multe roci.

Cerință

Determinați numărul maxim de roci pentru care exista cel puțin un traseu pe care îl poate urma robotul pentru a le preleva.

Date de intrare

Fișierul *marte.in* are următoarea structură:

- pe prima linie numerele N, E, P, X, Y separate prin câte un spațiu reprezentând dimensiunile matricii, numărul inițial de unități de energie, numărul de roci și perechea de coordonate din care pornește robotul.
- pe următoarele N linii, N șiruri de câte N valori întregi separate prin câte un spațiu reprezentând înălțimile celulelor din matrice.
- pe următoarele P linii perechi de coordonate A, B separate prin câte un spațiu, reprezentând coordonatele câte unei roci.

Date de ieșire

Fișierul *marte.out* va conține pe prima linie numărul de roci ce pot fi prelevate NR.

Restricții și precizări

- $1 \leq N \leq 100$, $0 \leq E \leq 1000$, $1 \leq P \leq 100$, $1 \leq A[I][J] \leq 50$
- toate valorile sunt numere întregi iar coordonatele celulelor cu roci respecta cerință de a fi în matrice.

Exemplu

| marte.in | marte.out | Explicatie |
|---|-----------|--|
| 4 3 3 2 2 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 2 3 4 4 1 1 | 2 | Robotul poate preleva rocile din celulele de coordonate (1,1) si (2, 3). |

Timp de executare: 0.1 secunde/test

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

Dimensiunea maximă a sursei: 10 KB

Secțiunea 9-10 începători

PROBLEMA 3 NUMERE

100 puncte

În vacanța impusă de restricțiile Covid-19 părinții lui Alex îi lasă în fiecare dimineață câte o provocare.

Astăzi el a primit o foie pe care se afla n numere naturale. Provocarea pe care o are este să reconstituie șirul inițial de $n+1$ numere naturale din care părinții săi au format șirul primit de el, aplicând formula de calcul a celui mai mare divizor comun pentru fiecare pereche de numere aflate pe poziții consecutive.

Cerință

Pentru un sir de n numere naturale $A(a_1, a_2, \dots, a_n)$ determinați șirul inițial $B(b_1, b_2, \dots, b_{n+1})$ din care părinții lui Alex au format șirul A . În cazul în care sunt mai multe variante de alegere a șirului B , să se determine șirul a cărui sumă a elementelor este minimă.

Date de intrare

Fișierul de intrare *numere.in* conține un șir de numere naturale A , separate prin câte un spațiu.

Date de ieșire

Fișierul de intrare *numere.out* conține pe prima linie numerele șirului B separate prin câte un spațiu iar pe a doua linie suma acestora.

Restricții și precizări

$1 < n \leq 1000$ pentru 70% din teste

$1 < n \leq 10^6$ pentru alte 30% din teste

$1 < a_i < 32800$

Exemplu

| numere.in | numere.out | Explicație |
|-----------|---------------|--|
| 1 2 3 | 1 2 6 3 12 | $\text{cmmdc}(1,2)=1$, $\text{cmmdc}(2,6)=2$, $\text{cmmdc}(6,3)=3$ Alt șirur posibil este 1 4 6 3 dar are suma elementelor mai mare ca 1 2 6 3 |

Timp maxim de execuție: 1 secundă/test

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

Dimensiunea maximă a sursei: 5 Kb

PROBLEMA 4 ROBO

100 puncte

Robo are de codificat de Crăciun cadourile pe care trebuie să le distribuie copiilor din ținutul Creo. El are la dispoziție n cifre de 0,1 sau 2 și p cutii de jucării. Robo trebuie să grupeze cifrele astfel încât să formeze cât mai multe coduri perfecte. Un cod este “perfect” dacă conține cel puțin o cifră de 1, cel puțin o cifră de 2 și este format din exact 3 cifre. Sunt cifre care nu aparțin nici unui cod și o cifră aparține unui singur cod. Cutiile de jucării sunt asociate unui cod perfect. Un cod perfect conține unul sau două cutii de jucării.

Secțiunea 9-10 începători

Cerință

Deoarece Robo dorește să reducă la minimum cheltuielile, este interesat să afle cutia cu cea mai mică dimensiune pe care o poate folosi pentru a-și împacheta întreaga colecție. Pentru ambalare se folosește aceeași dimensiune pentru fiecare cutie.

Date de intrare

Fișierul **robo.in** conține următoarea structură: pe prima linie sunt n, p două numere naturale.

Pe linia a doua linie sunt n cifre de 0,1 sau 2 separate printr-un spațiu.

Pe linia a treia linie sunt p numere naturale reprezentând dimensiunile cutiilor de jucării.

Date de ieșire

Fișierul **robo.out** conține pe prima linie numărul maxim de coduri pe care Robo le poate forma.

Pe linia a doua a fișierului dimensiunea minimă a cutiei de ambalare.

Restricții și precizări

- într-o cutie de ambalare de dimensiune s se poate cuprinde una sau două cutii de jucării dacă suma dimensiunilor lor nu depășește s
- $n \leq 250000$
- $p \leq 10000$
- $p \leq 2 \times \text{numărul maxim de coduri pe care Robo le poate forma}$
- dimensiunile cutiilor de jucării ≤ 20000

Exemplu

| robo.in | robo.out | Explicație |
|--|----------|--|
| 10 4 1 2 1 2 0 1 0 0 0 0 2 4 5 9 | 2 11 | Sunt 10 cifre si 4 cadouri. Cifrele se grupează în 2 coduri perfecte. Cifrele de pe pozițiile 1,2,5 formează primul cod perfect si al doilea cod perfect este format din cifrele de pe pozițiile 3,4, 7. Cifrele de pe pozițiile 6,8,9 si 10 nu sunt folosite. Cutiile de jucării sunt distribuite la 2 coduri astfel {1,4} si {2,3}. Dimensiunea cutiilor necesare este 11. |
| 10 4 1 2 1 2 0 1 2 0 1 1 2 4 5 9 | 3 9 | Sunt 3 coduri perfecte. O grupare posibila pentru a obține 3 coduri perfecte este {1,2,5}, {3,4,8} {6,7,9} Cele 4 cutii de jucării sunt ambalate in 3 cutii de dimensiune 9 |

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