

### V-uri

100 puncte

Se consideră un tablou bidimensional cu  $m$  linii și  $n$  coloane. Se numește traseu în V o parcurgere prin elementele tabloului astfel:

- se pleacă întotdeauna dintr-un element de pe linia 1 a tabloului, se ajunge în final într-un alt element de pe linia 1 a tabloului, trecând prin cel puțin 3 elemente, fără a trece printr-un element de mai multe ori;
- parcurgerea elementelor tabloului se face în forma unei singure litere V ca în desen, dintr-un element putându-se trece doar într-un alt element imediat vecin pe diagonală.

Fiecare element al tabloului conține valori întregi. La parcurgerea traseului se calculează suma elementelor de pe traseu.

### Cerință

Determinați traseul în V care conține cea mai mare sumă. În cazul în care există mai multe trasee cu aceeași sumă, se va alege traseul care parcurge cele mai puține celule. Dacă și în acest caz există mai multe soluții, se alege traseul cel mai din stânga (cel cu indicele coloanei de pornire cel mai mic).

### Date de intrare

Din fișierul de intrare **v.in** se citesc:

- de pe prima linie două valori naturale  $m$  și  $n$ , separate printr-un spațiu, reprezentând numărul de linii și numărul de coloane ale tabloului
- pe următoarele  $m$  linii se află valorile elementelor tabloului de pe fiecare linie, valori separate pe fiecare linie, două câte două, printr-un spațiu

### Date de ieșire

Fișierul de ieșire **v.out** va conține o singură linie pe care se află trei valori naturale reprezentând suma valorilor din elementele tabloului pentru traseul ales, coloana de pornire și linia pe care se află vârful V-ului.

**Atenție!** Datele din fișierul de ieșire trebuie să fie în ordinea specificată mai sus (sumă, coloană, linie).

### Restricții

- $1 \leq m, n \leq 100$
- $-60000 \leq$  valorile elementelor tabloului  $\leq 60000$

Conform datelor de intrare, suma valorilor din elementele oricărui traseu nu depășește 1.000.000.000

### Exemplu:

<b>v.in</b>	<b>v.out</b>
5 9	54 1 3
3 4 12 4 6 7 9 5 12	
0 4 5 7 9 -5 1 1 5	
0 98 34 0 1 7 7 1 1	
6 7 8 -9 0 2 3 5 22	
47 62 31 55 0 83 23 77 10	

Exemple de alte trasee care se pot crea conform datelor de mai sus, dar care au suma mai mică decât cea din rezultat:

3	4	12	4	6	7	9	5	12
0	4	5	9	9	-5	1	1	5
0	98	34	0	1	7	7	1	1
6	7	8	-9	0	2	3	5	22
47	62	31	55	0	83	23	77	10

**Timp maxim de execuție: 1 secundă/test**

### Jetoane

**100 puncte**

Ionel este elev în clasa a IV-a și părinții s-au gândit la o metodă eficientă prin care el să învețe numerele mari și modul lor de formare. În acest scop i-au cumpărat un joc cu jetoane. Pentru fiecare cifră există câte 10 jetoane inscripționate pe una din fețe cu cifra respectivă. Ionel poate forma numere prin așezarea jetoanelor unul lângă altul. Părinții îi cer lui Ionel să formeze numere, pe rând, care să aibă suma cifrelor  $S$  și să fie mai mici decât  $10^a$ . Pentru că este dificil să-l urmărească în formarea numerelor, părinții doresc să știe câte astfel de numere distincte știe Ionel să formeze.

### Cerință

Câte numere distincte a format Ionel?

### Date de intrare

În fișierul **jetoane.in**, pe prima linie se găsește numărul  $S$  iar pe a doua linie numărul  $a$ .

### Date de ieșire

În fișierul **jetoane.out** se va scrie pe un singur rând numărul cerut.

### Restricții

$$1 \leq S < 10$$

$$1 \leq a < 10$$

$S$  și  $a$  sunt numere naturale.

### Exemple

<b>jetoane.in</b>	<b>jetoane.out</b>
2 3	6
<b>jetoane.in</b>	<b>jetoane.out</b>
5 5	126

### Explicații

În cazul primului exemplu ( $S=2$  și  $a=3$ ) Ionel formează numerele: 2, 11, 20, 101, 110, 200.

**Timp maxim de execuție: 1 secundă/test**

### Neo

**100 puncte**

Neo, eroul din filmul Matrix, pentru a-și învinge adversarii are nevoie de energie. El a descoperit în matrice o zonă care cuprinde locații consecutive numerotate de la  $1$  la  $N$ , fiecare dintre ele generând o anumită cantitate de energie, pozitivă sau negativă, cunoscută.

Neo are posibilitatea să pătrundă în această zonă printr-o singură locație și poate părăsi zona o singură dată. El, odată intrat în zona energetică, trebuie să străbată locațiile în ordinea crescătoare a numărului lor, iar energia acumulată într-o locație multiplică de atâtea ori energia deja acumulată de

Neo. De asemenea, dacă într-o locație se află energie negativă, iar Neo a acumulat până în acea locație energie pozitivă, toată energia lui se transformă în energie negativă, dar dacă în drumul său Neo întâlnește o altă locație cu energie negativă, iar energia deținută de el este tot negativă, toată energia lui devine pozitivă. Energia inițială a lui Neo are valoarea 1.

### **Cerință**

Ajutați-l pe Neo să descopere locația în care trebuie să intre și locația în care trebuie să părăsească zona energetică pentru a acumula cea mai mare cantitate de energie pozitivă posibilă.

### **Date de intrare**

Din fișierul **NEO.IN** se citesc:

- de pe linia 1 numărul locațiilor energetice
- de pe linia 2 un șir de numere întregi reprezentând cantitatea de energie a fiecărei locații, valori separate două câte două printr-un spațiu

### **Date de ieșire**

În fișierul **NEO.OUT** se scriu:

- pe linia 1 cantitatea de energie acumulată de NEO
- pe linia 2 două valori separate printr-un spațiu: numărul de ordine al locației în care NEO pătrunde în zona energetică și numărul de ordine al locației în care NEO iese din zona energetică. Dacă nu se poate obține energie pozitivă, cantitatea de energie acumulată și numărul de ordine al locațiilor vor fi 0.

### **Restricții și precizări:**

$1 \leq N \leq 100$ ,

$-3 \leq$  valoarea energiei dintr-o locație  $\leq 3$

$-2.000.000.000 \leq$  produsul tuturor energiilor  $\leq 2.000.000.000$

Dacă există mai multe soluții pentru datele de intrare, se va afișa una singură.

### **Exemple:**

<b>NEO . IN</b>	<b>NEO . OUT</b>
4	1 2
-3 -2 2 -3	2 4
<b>NEO . IN</b>	<b>NEO . OUT</b>
1	0
-2	0 0

**Timp maxim de execuție/test: 1 secundă.**