

## Problema 1 – domino

100 de puncte

Într-un joc de domino, fiecare piesă este împărțită în două zone, în fiecare zonă fiind înscris un număr natural. Dacă jocul are dimensiunea  $d$ , în joc vor exista toate piesele distincte care se pot forma cu numere cuprinse între 0 și  $d$ . Două piese sunt considerate *identice* dacă au înscrise aceleași numere, indiferent de ordinea lor. Astfel, piesele  $(3, 7)$  și  $(7, 3)$  sunt identice. De exemplu, jocul de dimensiune  $d=2$  va avea 6 piese distincte:

0	0	0	1	0	2	1	1	1	2	2	2
---	---	---	---	---	---	---	---	---	---	---	---

Suma tuturor numerelor de pe aceste piese este 12. Problema are două cerințe:

- Dat fiind un șir format din  $N$  numere naturale nenule reprezentând dimensiunile unor jocuri de domino, să se determine pentru fiecare joc suma tuturor numerelor înscrise pe piesele din jocul respectiv.
- Dat fiind un șir format din  $N$  numere naturale nenule reprezentând sumele tuturor numerelor de pe piesele unor jocuri de domino, se construiește mai întâi un șir de cifre, notat cu  $A$ , scriind în ordine toate numerele din șirul dat, fără spații între ele. Se cere să se construiască un șir strict crescător de numere naturale, notat cu  $B$ , parcurgând alternativ cifrele din șirul  $A$  de la stânga la dreapta și de la dreapta la stânga după cum urmează:
  - primul număr din  $B$  este format din prima cifră din șirul  $A$ ;
  - al doilea număr din  $B$  se construiește concatenând (alipind) cifrele din  $A$ , începând de la dreapta către stânga, până când obținem un număr strict mai mare decât primul număr din  $B$ ;
  - al treilea număr din  $B$  se construiește concatenând cifrele din  $A$  de la stânga către dreapta (începând cu prima cifră care nu a fost deja utilizată), până când obținem un număr strict mai mare decât precedentul din  $B$ ;
  - al patrulea număr din  $B$  se construiește concatenând din nou cifrele din  $A$  de la dreapta la stânga (începând cu cea mai din dreapta cifră care nu a fost deja utilizată), până când obținem un număr strict mai mare decât al treilea din  $B$ ;
  - se continuă astfel alternativ, până când nu se mai poate forma un număr strict mai mare decât ultimul număr adăugat în  $B$ .

### Cerință

Scrieți un program care rezolvă cerințele 1 și 2 descrise în enunț.

### Date de intrare

Fișierul de intrare `domino.in` conține pe prima linie un număr natural  $C$  reprezentând cerința care trebuie rezolvată (1 sau 2). Pe a doua linie se află numărul natural  $N$ . Pe a treia linie se află  $N$  numere naturale nenule separate prin câte un spațiu  $d_1 d_2 \dots d_N$ .

### Date de ieșire

Fișierul de ieșire `domino.out` va conține o singură linie. Dacă  $C=1$ , pe prima linie se vor afișa  $N$  numere naturale separate prin câte un spațiu; al  $i$ -lea număr afișat reprezintă suma numerelor din jocul de domino având dimensiunea  $d_i$  ( $1 \leq i \leq N$ ). Dacă  $C=2$ , pe prima linie se vor afișa în ordine, separate prin câte un spațiu, valorile din șirul  $B$  determinat conform regulilor din enunț.

### Restricții și precizări

- $1 \leq N \leq 10^4$
- Dacă  $C=1$ ,  $1 \leq d_i \leq 1000$ , iar dacă  $C=2$ ,  $1 \leq d_i \leq 10^9$ , pentru  $1 \leq i \leq N$ .
- Numerele din șirul  $B$  vor fi afișate fără zerouri nesemnificative (de exemplu, dacă în urma aplicării regulilor din enunț în șirul  $B$  se obține numărul 0204 se afișează 204).
- Pentru teste în valoare de 30 de puncte cerința este 1.

### Exemple

<code>domino.in</code>	<code>domino.out</code>	Explicație																					
1 5 2 3 15 4 7	12 30 2040 60 252	Cerința este 1, deci trebuie să determinăm sumele numerelor din jocurile de dimensiune 2, 3, 15, 4 și 7.																					
<code>domino.in</code>	<code>domino.out</code>	Explicație																					
2 5 12 30 2040 60 252	1 2 23 52 204	Din șirul 12 30 2040 60 252 se formează: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Șirul A</th> <th>Șirul B</th> <th>Sensul de formare</th> </tr> </thead> <tbody> <tr> <td><del>1</del>230204060252</td> <td>1</td> <td>→</td> </tr> <tr> <td>23020406025<del>2</del></td> <td>1 2</td> <td>←</td> </tr> <tr> <td><del>2</del>3020406025</td> <td>1 2 23</td> <td>→</td> </tr> <tr> <td>0204060<del>2</del>5</td> <td>1 2 23 52</td> <td>←</td> </tr> <tr> <td><del>0</del>204060</td> <td>1 2 23 52 204</td> <td>→</td> </tr> <tr> <td>060</td> <td>nu se mai poate forma un număr &gt; 204.</td> <td></td> </tr> </tbody> </table>	Șirul A	Șirul B	Sensul de formare	<del>1</del> 230204060252	1	→	23020406025 <del>2</del>	1 2	←	<del>2</del> 3020406025	1 2 23	→	0204060 <del>2</del> 5	1 2 23 52	←	<del>0</del> 204060	1 2 23 52 204	→	060	nu se mai poate forma un număr > 204.	
Șirul A	Șirul B	Sensul de formare																					
<del>1</del> 230204060252	1	→																					
23020406025 <del>2</del>	1 2	←																					
<del>2</del> 3020406025	1 2 23	→																					
0204060 <del>2</del> 5	1 2 23 52	←																					
<del>0</del> 204060	1 2 23 52 204	→																					
060	nu se mai poate forma un număr > 204.																						

**Timp maxim de executare: 0.2 secunde**

**Total memorie disponibilă 8 MB din care pentru stivă 4 MB**

**Dimensiunea maximă a sursei: 10 KB**

**Sursa: `domino.cpp`, `domino.c` sau `domino.pas` va fi salvată în folderul care are drept nume ID-ul tău.**