

**PROBLEMA 1      Artefacte****100 puncte**

Aventurierul celebru și neînfricat, Mihăiță ajuns într-un templu, intră într-o cameră labirint. La intrarea acesteia, se află inscripționată pe un perete harta acesteia, o matricea binară pătratică ce reprezintă labirintul (0-spații libere, 1-pereți) și se menționează că ușa de ieșire poate fi deschisă cu ajutorul unei chei care este alcătuită din trei artefacte împrăștiate prin labirint.

**Cerința**

Cunoscând dimensiunea labirintului  $N$ , matricea binară  $N \times N$  asociată acestuia și locațiile artefactelor. Mihăiță vă roagă să găsiți lungimea minimă a traseului pe care trebuie să îl parcurgă pentru a colecta cele 3 artefacte și a ieși din labirint.

**Date de intrare**

Pe prima linie a fișierului *artefacte.in* se află  $N$  - dimensiunea matricii. Pe următoarele  $N$  linii se află harta labirintului, fiecare linie are valori de 0 și 1 separate printr-un spațiu. Pe următoarele 3 linii se află câte 2 numere care reprezintă coordonatele celor 3 artefacte, în ordinea linie, coloană, separate prin câte spațiu.

**Date de ieșire**

Pe prima linie a fișierului *artefacte.out* se află lungimea minimă cerută.

**Restricții și precizări**

- $3 \leq N \leq 35$
- Locațiile artefactelor sunt unice
- Intrarea în cameră este identificată prin poziția (1,1) în matrice și ieșirea din ea este identificată prin poziția (N,N) în matrice, iar deplasarea nu se poate face decât în 4 direcții (N,S,E,V)
- Se garantează existența unei soluții

**Exemplu**

artefacte.in	artefacte.out	Explicație
4 0 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 3 3 1 4 3 1	12	Traseul de lungime minimă este: de la (1,1) la (1,4) (lungime 3) se colectează un artefact de la (1,4) la (3,1) (lungime 5) se colectează un artefact de la (3,1) la (3,3) (lungime 2) se colectează un artefact de la (3,3) la (4,4) (lungime 2) Lungime totală 12

**PROBLEMA 2      Wii****100 puncte**

Mimirdon tocmai și-a cumpărat un Nintendo Wii. Nintendo-ul său este însă unul mai special: conține  $n$  jocuri, care trebuie neapărat jucate în ordine; dacă acum se joacă jocul numărul  $i$ , următorul joc va fi jocul  $i+1$  sau 1 în cazul în care  $i=n$ .

## Secțiunea 7-8 avansați

Interesant e că fiecare joc are asociat un grad de distracție  $x$  și pentru orice sesiune de gaming a lui Mimirdon, jocul următor nu poate fi jucat decât dacă are gradul de distracție cel puțin egal cu jumătate din maximul gradelor de distracție ale jocurile jucate până la acel moment în sesiunea curentă. Sesiunea de gaming se oprește atunci când aceasta condiție nu mai este îndeplinită sau când Mimirdon, plictisit, va decide oprirea jocului (există și situații în care jocul nu se va opri).

**Cerința**

Știind că sesiunea de gaming poate începe cu oricare din cele  $n$  jocuri, determinați cât de lungă va fi sesiunea de joc a lui Mimirdon (câte jocuri va juca) sau dacă Mimirdon, plictisit, va decide oprirea jocului.

**Date de intrare**

Fișierul *wii.in* conține pe prima linie un număr natural nenul  $n$ , reprezentând numărul de jocuri din Wii-ul lui Mimirdon. Următoarea linie conține  $n$  numere naturale nenule, separate prin câte un spațiu, reprezentând gradele de distracție ale celor  $n$  jocuri.

**Date de ieșire**

Fișierul *wii.out* va conține pe prima linie numerele  $x_1, x_2, \dots, x_n$ , separate prin câte un spațiu, fiecare număr  $x_i$  reprezentând câte jocuri va juca Mimirdon dacă începe sesiunea de pe poziția  $i$  sau  $-1$  dacă Mimirdon este cel care va decide oprirea sesiunii **deoarece aceasta ar continua la infinit**.

**Restricții și precizări**

- $2 \leq n \leq 100000$ .
- $1 \leq x_i \leq 10^9$ , unde  $x_i$  reprezintă gradele de distracție ale jocurilor.

**Exemple**

wii.in	wii.out	Explicații
6 6 8 4 2 5 1	3 2 3 2 1 4	Dacă Mimirdon începe sesiunea de gaming cu jocul cu numărul: <ul style="list-style-type: none"> <li>• 1, atunci acesta se va opri după 3 jocuri, deoarece al 4-lea joc are gradul 2 care nu este cel puțin egal cu jumătatea maximului (<math>8/2</math>) dintre jocurile parcurse.</li> <li>• 2, atunci acesta se va opri după 2 jocuri, căci <math>2 &lt; 8/2</math>.</li> <li>• 3, atunci acesta se va opri după 3 jocuri, căci <math>1 &lt; 5/2</math>.</li> <li>• 4, atunci acesta se va opri după 2 jocuri, căci <math>1 &lt; 5/2</math>.</li> <li>• 5, atunci acesta se va opri după 1 joc, căci <math>1 &lt; 5/2</math>.</li> <li>• 6, atunci acesta se va opri după 4 jocuri, căci <math>2 &lt; 8/2</math>.</li> </ul>
5 6 8 5 7 4	-1 -1 -1 -1 -1	De oriunde ar începe Mimirdon, acesta va decide oprirea jocului.
5 3 3 2 1 3	3 2 6 5 4	Dacă Mimirdon începe sesiunea de gaming cu jocul cu numărul: <ul style="list-style-type: none"> <li>• 1, atunci acesta se va opri după 3 jocuri, căci <math>1 &lt; 3/2</math>.</li> <li>• 2, atunci acesta se va opri după 2 jocuri, căci <math>1 &lt; 3/2</math>.</li> <li>• 3, atunci acesta se va opri după 6 jocuri, căci <math>1 &lt; 3/2</math>.</li> <li>• 4, atunci acesta se va opri după 5 jocuri, căci <math>1 &lt; 3/2</math>.</li> <li>• 5, atunci acesta se va opri după 4 jocuri, căci <math>1 &lt; 3/2</math>.</li> </ul>

**PROBLEMA 3 Pinguini****100 puncte**

Mihăiță s-a implicat în studierea pinguinilor extraterestrii și a modului lor de acțiune. El a observat că navele pinguinilor au diferite ranguri, că există pinguini pașnici dar și pinguini ostili care doresc să distrugă orașul Ploiești și că pinguinii vor să atace în formație. Pentru simplitate el consideră planeta un plan cartezian pe care notează locurile de aterizare ale navelor și încearcă să determine formațiile ostile.

4 nave pot forma o formație ostilă, dacă dreptunghiul format de punctele lor de aterizare

- are laturile paralele cu axele OX,OY
- cele 4 nave au același rang

**Cerința**

Mihăiță vă roagă să scrieți un program care să-l ajute să numere formațiile ostile.

**Date de intrare**

Pe prima linie a fișierului de intrare *pinguini.in* se află două numere naturale  $N$  – numărul de nave și  $nrang$  – rangul maxim pe care îl poate avea o nava. Pe următoarele  $N$  linii se află câte 3 numere  $X, Y$  coordonatele locului de aterizare în ordinea axa  $Ox$ , axa  $Oy$  și rangul navei  $C$ .

**Date de ieșire**

În fișierul *pinguini.out*, prima linie conține numărul de formații ostile.

**Restricții și precizări**

- $1 \leq N \leq 1000$
- $1 \leq nrang \leq 5$
- $-1000 \leq X, Z \leq 1000$
- Pentru 40% din teste  $N \leq 100$
- Se garantează că testele au soluții
- Punctele de aterizare sunt unice

**Exemplu**

<b>pinguini.in</b>	<b>pinguini.out</b>	<b>Explicație</b>
9 1 0 0 1 1 1 1 2 2 1 0 1 1 1 0 1 0 2 1 2 0 1 1 2 1 2 1 1	9	Se pot forma 9 dreptunghiuri cu rangul 1 identificate prin colțurile stânga sus, dreapta jos astfel (0,2),(1,1) (0,1),(1,0) (1,2),(2,1) (1,1),(2,0) (0,2),(2,1) (0,1),(2,0) (0,2),(1,0) (1,2),(2,0) (0,2),(2,0)

**PROBLEMA 4 Parola****100 puncte**

În ultima vreme Mimir s-a jucat prea mult, așa că mama lui a decis să îi pună o parolă pe laptop. Știind că Mimir va încerca să afle parola l-a "provocat" cu o problemă de șiruri.

"Parola este numărul de subșiruri bune dintr-un șir de caractere alcătuit doar din caracterele **D** și **E**. Un **subșir bun** este un subșir în care fiecare caracter face parte din cel puțin un palindrom".

De exemplu, pentru șirul DEEDDD, EEDDD este un subșir bun, deoarece fiecare caracter face parte din cel puțin un palindrom (EE este un palindrom, dar și DDD este este palindrom).

**Cerința**

Cunoscând șirul de caractere *s*, cu care mama lui Mimir a realizat parola, ajutați-l pe Mimir să descifreze parola și să se joace din nou.

**Date de intrare**

Pe primul rând al fișierului *parola.in* se află numărul *n*, reprezentând numărul de caractere al șirului *s*, iar pe al doilea rând se află șirul *s*.

**Date de ieșire**

Pe primul rând al fișierului *parola.out* se va afișa numărul de subșiruri bune. Deoarece numărul poate fi foarte mare, rezultatul va fi afișat modulo **1000000007**.

**Restricții și precizări**

- $1 < N \leq 300000$
- Un subșir bun are minim 2 caractere

**Exemplu**

<b>parola.in</b>	<b>parola.out</b>	<b>Explicație</b>
6 EEEDDD	10	Subșirurile bune sunt:  s1-s2 (EE), s2-s3 (EE), s1-s3 (EEE), s4-s5 (DD), s5-s6 (DD), s4-s6 (DDD), s1-s5 (EEEDD), s2-s5 (EEDD), s1-s6 (EEEDDD), s2-s6 (EEDDD)