

## Secțiunea 11-12

## PROBLEMA 1

100 puncte

Un număr  $X$  se numește oltenesc dacă nu conține nici o putere de 2 de cel puțin 2 cifre ca subsecvență în scrierea sa zecimală. Nea Mărin are un număr  $N$  format din cel mult 100 de cifre și se întreabă câte numere naturale cel mult egale cu  $N$  sunt oltenești. Deoarece răspunsul poate fi destul de mare, se cere doar restul împărțirii sale la  $10^9 + 7$ .

## Cerință

Se dau  $T$  întrebări, fiecare constând dintr-un singur număr  $N$ . Pentru fiecare întrebare să se calculeze câte numere  $0 \leq X \leq N$  sunt oltenești, modulo  $10^9 + 7$ .

## Date de intrare

Pe prima linie a fișierului *oltenesc.in* se află numărul  $T$ . Urmează  $T$  linii, fiecare conținând câte un număr  $N$  format din cel mult 100 de cifre zecimale, reprezentând o întrebare.

## Date de ieșire

Fișierul de ieșire *oltenesc.out* va conține  $T$  linii, constând în răspunsurile la cele  $T$  întrebări din fișierul de intrare.

## Restricții și precizări

- $1 \leq T \leq 10$ ;
- $1 \leq N < 10^{100}$ ;
- Pentru 10% din teste,  $N < 10^6$ ;
- Pentru 50% din teste,  $N < 10^{18}$ .
- Prin subsecvență a unui număr zecimal înțelegem numărul obținut prin păstrarea unei secvențe continue formate din cifrele numărului inițial. De exemplu, 234, 12 și 12345 sunt subsecvențe ale numărului 12345, dar 135 și 432 nu sunt.

## Exemplu

oltenesc.in	oltenesc.out	Explicație
4	32	Singurele numere care nu sunt oltenești cuprinse între 0 și 33 sunt 16 și 32. Singurele numere care nu sunt oltenești cuprinse între 0 și 999 sunt 128, 256, 512 și cele de formele $16?$ , $32?$ , $64?$ , $?16$ , $?32$ , $?64$ (observați cum numărul 164 se regăsește de 2 ori în această listă).
33	938	
999	194003	
232323	515744048	
992391662939123897		

**Timp maxim de execuție:** 2 secunde/test

**Memorie totală disponibilă:** 512 MB din care 512 MB pentru stivă

**Dimensiunea maximă a sursei:** 10 Kb



## Secțiunea 11-12

## PROBLEMA 2 CARACATIȚĂ

100 puncte

Observăm că șirurile a, abb, cdd, opp, xzzyyy, qppaaammm au ceva în comun. Mai exact, primul caracter se repetă o dată, al doilea de 2 ori, al treilea de 3 ori, etc. Vom numi aceste șiruri vrednice. Se dau N șiruri vrednice de caractere.

**Cerință**

Se dau M query-uri. Fiecare query este un șir de caractere. Vrem să aflăm de câte ori fiecare query se “potrivește” în cele N șiruri date. Considerăm că un query q se potrivește într-un șir s dacă:

- q este subsir al lui s (abc este subsir al lui axbxcx)
- dacă asupra lui q și s se aplică o operație care restrânge toate caracterele adiacente egale într-unul singur, q este subsecvență a lui s (abc este subsecvență a lui xabcx) (q se potrivește în s de câte ori q restrâns apare ca subsecvență în s restrâns)

De exemplu, abc se potrivește în abbccc, xaabbccc o dată, în abbcccaaaabbbbcccc de 2 ori, și în abbbcccc, axxbbcccc, abb niciodată.

**Date de intrare**

Pe prima linie a fișierului *caracatita.in* se află numărul N. Urmează N linii care conțin fiecare câte un șir vrednic de caractere, conținând doar caracterele a-z.

Pe următoarea linie se afla numărul M. Urmează M linii care conțin fiecare câte un șir de caractere conținând doar caracterele a-z.

**Date de ieșire**

Pe linia i din fișierul de ieșire *caracatita.out* se va afla numărul de potriviri al query-ului i.

**Restricții și precizări**

- $1 \leq N \leq 500$
- $1 \leq \text{Lungimea unui șir vrednic} \leq 5050$
- $1 \leq M \leq 100\,000$
- $2 \leq \text{Lungimea unui șir query} \leq 1000$
- Pentru 20 % din teste,  $N \leq 50$
- Pentru alte 20 % din teste,  $M \leq 1000$

**Exemple**

caracatita.in	caracatita.out
1	1
abbccddddd	1
4	1
abc	0
abcc	
abccc	
abcccc	

## Secțiunea 11-12

caracatita.in	caracatita.out
2	0
axxyyyzzzzppppp	0
xyy	1
5	2
axp	1
axz	
xxyyzzz	
xyy	
z	

**Timp maxim de execuție:** 0.5 secunde/test

**Memorie totală disponibilă:** 300 MB, din care 300 MB pentru stivă

**Dimensiunea maximă a sursei:** 10 Kb

**PROBLEMA 3 DOUBLEDROP****100 puncte**

3XIMO este un DJ faimos. El vrea să pună o secvență de melodii 2 câte 2 (double drop) la o anumită petrecere.

O n-melodie este o secvență de n vibrații(V) și n pauze (P) ( $n \geq 0$ ).

O n-melodie k-rupe dacă numărul de prefixe care se termină în V, cu diferență dintre V și P strict pozitivă, este egală cu k.

De exemplu: (ultima literă dintr-un prefix care contribuie la k-rupere este îngroșată):

- **VVVPVVP** este o 5-melodie care 5-rupe
- **VPVPPV** este o 4-melodie care 2-rupe
- **PPVV** este o 2-melodie care 0-rupe

3XIMO își cunoaște foarte bine publicul și știe la fiecare moment i următoarele informații:

$N[i]$   $M[i]$  - publicul dorește ca 3XIMO fie să facă double drop (ceea ce înseamnă că alege 2 melodii și le pune într-o ordine) cu 2 melodii într-o ordine: prima o j-melodie care  $\min(j, M[i])$ -rupe și a doua o  $(N[i]-j)$ -melodie care  $\min((N[i]-j), M[i])$ -rupe. ( $0 \leq j \leq N[i]$ ,  $M[i] \leq N[i]$ ,  $1 \leq i \leq LEN$ )

Deci la momentul i, 3XIMO poate alege oricare 2 melodii care satisfac constrângerile.

O secvență de double drop-uri se numește “blanaos” dacă respectă toate informațiile la fiecare moment. 3XIMO vă roagă să îi spuneți câte secvențe “blanaos” sunt modulo  $10^9 + 7$ .

**Date de intrare**

Fișierul *doubledrop.in* conține pe prima linie un număr LEN, egal cu lungimea secvenței de double dropuri. Pe a (i+1)-a linie vor apărea  $N[i] \leq 1e5$  și  $M[i] \leq 1e5$  cu semnificația din problema.

**Date de ieșire**

Fișierul *doubledrop.out* conține pe prima linie, numărul de secvențe “blanaos” modulo  $10^9 + 7$ .

**Restricții și precizări**

- $LEN \leq 10$ ,  $0 \leq M[i] \leq N[i] \leq 10$  pentru 10 puncte;
- $LEN \leq 4000$ ,  $0 \leq M[i] \leq N[i] \leq 4000$  pentru alte 30 de puncte;

## Secțiunea 11-12

- $LEN \leq 100.000$ ,  $0 \leq M[i] \leq N[i] \leq 100.000$  pentru 100 de puncte;
- Prin convenție, exista o singura 0-melodie care 0-rupe;

## Exemple

doubledrop.in	doubledrop.out	Explicație
2 1 1 0 0	2	3XIMO la timp 1 poate să facă double drop în următoarele moduri: Prima o 1-melodie care 1-rupe (există doar o astfel de melodie) cu a doua o 0-melodie care 0-rupe (există doar o astfel de melodie) Prima o 0-melodie care 0-rupe (există doar o astfel de melodie) cu a doua o 1-melodie care 1-rupe (există doar o astfel de melodie) 3XIMO la timp 2 poate să facă double drop în următorul mod: Prima o 0 melodie care 0-rupe (există doar o astfel de melodie) și a doua același fel de melodie Deci 3XIMO în total poate alege o subsecvență de double drop-uri "blanaos" în $(1+1)*1 = 2$ moduri
4 0 0 0 0 0 0 0 0	1	Există o posibilitate pentru fiecare drop deci există o singură secvență de double drop-uri "blanaos".
3 2 1 0 0 3 2	70	

**Timp maxim de execuție:** 0.5 secunde/test

**Memorie totală disponibilă:** 16 MB din care 16 MB pentru stivă

**Dimensiunea maximă a sursei:** 5 Kb

**PROBLEMA 4 LANȚURI****100 puncte**

Se dă un arbore oarecare (graf neorientat conex și fără cicluri) cu  $n$  noduri. Fiecare nod are asociată o cheie (număr natural). Determinați numărul de lanțuri elementare care au suma cheilor asociate nodurilor componente un număr divizibil cu 3. Un lanț se cheamă elementar dacă nodurile sale sunt distincte. Două lanțuri se consideră distincte dacă diferă prin cel puțin un nod sau prin ordinea de așezare a nodurilor. Un lanț se consideră identic cu cel obținut citind nodurile invers.

**Date de intrare**

Fișierul *lanturi.in* conține pe prima linie un număr natural  $n$  – numărul de noduri ale arborelui. Pe linia a doua sunt valorile cheilor nodurilor, separate prin câte un spațiu. Pe fiecare din următoarele

## Secțiunea 11-12

n-1 linii se află câte două numere naturale separate prin spațiu, reprezentând capetele câte unei muchii.

**Date de ieșire**

Fișierul *lanturi.out* conține restul împărțirii valorii determinate la 10007 (zece mii șapte).

**Restricții și precizări:**

- $2 \leq n \leq 100000$ ;
- Cheile sunt numere naturale de maxim 3 cifre;
- Nodurile sunt numerotate de la 1 la n.
- Considerăm inclusiv lanțurile de lungime 0 (cele formate dintr-un singur nod).

**Exemplu**

lanturi.in	lanturi.out	Explicație
4 1 2 3 4 1 2 1 3 4 1	3	Cele 3 lanțuri sunt: [3], [1,2], [3,1,2]

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

**Memorie totală disponibilă:** 10 MB, din care 10 MB pentru stivă

**Dimensiunea maximă a sursei:** 10 Kb