

**Ziua 1**  
**seti**

sursa: seti.pas, seti.cpp, seti.c

**Clasa a IX-a**

Cercetătorii ce lucrează la programul SETI au recepționat două transmisii de date foarte ciudate, date care ar putea veni din partea unor civilizații extraterestre. Primul set de date este format din 10 caractere distincte, date în ordinea lor lexicografică, ce formează alfabetul extraterestru. A doua transmisie conține cuvinte din exact 4 caractere.

**Cerință**

Cercetătorii trebuie să ordoneze lexicografic cuvintele primite în a doua transmisie (conform alfabetului extraterestru).

**Date de intrare**

Fișierul de intrare **seti.in** conține pe prima linie cele 10 caractere ale alfabetului, iar pe fiecare din următoarele linii câte un cuvânt.

**Date de ieșire**

Fișierul de ieșire **seti.out** va conține cuvintele ordonate, câte unul pe linie.

**Restricții**

În fișier nu sunt mai mult de 200.000 de cuvinte, iar caracterele sunt literele mici ale alfabetului englez.

Datele de intrare se presupun ca fiind corecte.

**Exemplu**

Pentru fișierul de intrare

**seti.in**

abcdefghijkl

aaaa

fgaa

aabc

iihf

Se obține fișierul de ieșire

**seti.out**

aaaa

aabc

fgaa

iihf

**Timp de execuție/test:** 1 secundă

## Ziua 1

sursa : scaune.pas, scaune.cpp, scaune.c

### scaune

**Clasa a IX-a**

Se consideră **ns** scaune numerotate de la 1 la **ns**, aranjate în cerc.

Exemplu pentru **ns=20** așezarea scaunelor este dată în figură.

	20	19	18	17	16	15	14	13	12	11
	<		<	<	<	<	<	<	<	<
	v									^
	>		>	>	>	>	>	>	>	>
	1	2	3	4	5	6	7	8	9	10

Pe fiecare din aceste scaune este așezat un copil. Primul copil stă pe scaunul 1, iar ultimul pe scaunul **ns**.

Pe lângă cele **ns** scaune deja ocupate, alți **nc** copii ( $1 \leq nc \leq ns$ ) așteaptă să se elibereze un loc.

La un moment dat un singur copil se ridică de pe scaun și pleacă. Atunci, cât timp în dreptul scaunului liber nu există un copil, toți copiii aflați în așteptare se mișcă în sens invers mișcării acelor ceasornicului, câte o poziție, până când unul din ei ocupă locul liber.

Condiții :

- la început toate scaunele sunt ocupate;
- fiecare copil aflat în așteptare se află inițial în dreptul unui scaun ocupat;
- când un copil avansează cu  $n$  poziții spre un loc pe scaun, toți cei care așteaptă avansează tot cu  $n$  poziții. Deoarece mișcarea este circulară, avansarea cu 4 poziții de la poziția 18, semnifică o deplasare în dreptul poziției 2;

### Cerință

Dacă se dă o secvență a numerelor de ordine a copiilor care pleacă la fiecare moment, să se scrie un program care să afișeze numărul scaunului pe care s-a așezat fiecare copil care așteaptă, dacă acest lucru este posibil.

### Date de intrare

Pe prima linie a fișierului de intrare **scaune.in** se află două numere, separate prin spațiu, reprezentând numărul de scaune, **ns** și respectiv numărul copiilor care stau în așteptare **nc**. Pe următoarele **nc** linii vor fi date pozițiile copiilor aflați în așteptare. În continuare până la sfârșitul fișierului sunt linii ce descriu numerele de ordine ale copiilor care se ridică unul câte unul de pe scaune și părăsesc jocul.

### Date de ieșire

Fișierul de ieșire **scaune.out** conține **nc** linii, fiecare linie conținând poziția inițială de așteptare a copilului și poziția ocupată, separate printr-un spațiu. Liniile de ieșire trebuie să fie în aceeași ordine ca cele din fișierul de intrare. În cazul în care un copil nu are nici o posibilitate să se așeze, în dreptul său se va scrie 0 în fișierul de ieșire.

### Restricții

- $1 \leq ns \leq 200$
- $nc \leq ns$

scaune.in	scaune.out
20 5	6 16
6	19 3
19	17 0
17	13 20
13	1 1
1	
1	
3	
20	
16	

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

## Ziua 1

sursa: `circular.pas`, `circular.cpp`, `circular.c`

## circular

## Clasa a IX-a

Unele numere naturale sunt formate doar din cifre distincte nenule. Dintre acestea, unele, numite **numere circulare**, au următoarea proprietate: pornind de la prima cifră și numărând spre dreapta, după cifră, atâtea cifre cât indică aceasta, se determină o nouă cifră. Procedând la fel și pentru aceasta și pentru toate cele care urmează se va ajunge din nou la prima cifră. Dacă toate cifrele au fost vizitate exact o dată, numărul se numește circular. De exemplu numărul

1894256

este număr circular deoarece:

- are numai cifre distincte
- nu conține cifra 0
- pornind de la 1 obținem, pe rând: 8, 9, 2, 6, 5, 4, 1

## Cerință

Scrieți un program care, pentru un  $N$  dat, determină câte numere circulare sunt mai mici sau egale cu  $N$ , precum și cel mai mare număr circular mai mic sau egal cu  $N$ .

## Date de intrare

Pe prima linie a fișierului de intrare **circular.in** se află numărul natural  $N$ .

## Date de ieșire

Fișierul de ieșire **circular.out** conține o singură linie, pe care se află numărul de numere circulare mai mici ca  $N$  precum și numărul circular maxim cerut, separate printr-un spațiu. Dacă nu există nici un număr circular mai mic ca  $N$ , în fișierul de ieșire se vor afișa două valori 0 separate printr-un spațiu.

## Restricții

- $10 \leq N < 10000000$

## Exemplu

<b>circular.in</b>	<b>circular.out</b>	<i>Semnificație</i>
1894250	347 1849625	Există 347 numere circulare mai mici ca 1894250 cel mai mare dintre acestea fiind 1849625

**Timp de execuție/test:** 1 secundă

## Ziua 2

sursa: criptare.pas, criptare.cpp, criptare.c

## criptare

## Clasa a IX-a

Mircea și Vasilică vor să-și trimită mesaje pe care alții să nu le înțeleagă. Au citit ei despre spioni și modalități de a scrie mesaje și, în final, au imaginat un mod de criptare a unui mesaj care folosește “cuvânt cheie” (le-a plăcut lor denumirea asta :-)).

Alegându-și un cuvânt cheie format numai din litere distincte, ei numără literele acestuia și împart mesajul în grupe de lungime egală cu numărul de litere ale cuvântului cheie, și le așează una sub alta. Desigur, se poate întâmpla ca ultima grupă să fie incompletă, așa că o completează cu spații. Apoi numerotează literele cuvântului cheie în ordinea apariției lor în alfabetul englez. În final, rescriu mesajul astfel: coloana de sub litera numerotată cu 1, urmată de coloana de sub litera numerotată cu 2, etc. înlocuind totodată și spațiile cu caracterul ‘\*’ (asterisc).

Exemplu:

cuvântul cheie	<b>criptam</b>
mesaj de criptat	Incercam sa lucram cu coduri si criptari.
cuvântul cheie	<b>criptam</b> are 7 litere
numerotare	<b>2635714</b>
deoarece, avem, în ordine	abcdefghijklmnopqrstu
	vwxyz
	1 2 3 4 5 6 7
împărțire în grupe	Incerca m sa lu cram cu  coduri  si cri ptari.
codificare	<b>2635714</b>
	Incerca
	m*sa*lu
	cram*cu
	*coduri
	*si*cri
	ptari.*
mesaj criptat	clcrr.Imc**pcsaioiaauuii*eamd*rn*rcstr**uci
	col1 col2 col3 col4 col5 col6 col7

## Cerință

Fiind date un cuvânt cheie și un mesaj criptat, decodificați mesajul trimis de Mircea pentru Vasilică.

## Date de intrare

Fișierul de intrare **criptare.in** conține pe prima linie mesajul criptat iar pe linia a doua cuvântul cheie.

## Date de ieșire

Fișierul de intrare **criptare.out** conține pe prima linie mesajul decriptat.

## Restricții

- lungimea mesajului este de minim 20 și maxim 1000 caractere
- cuvântul cheie are minim 5 și maxim 20 de caractere
- cuvântul cheie conține numai litere mici ale alfabetului

## Exemplu

<b>criptare.in</b>	<b>criptare.out</b>
clcrr.Imc**pcsaioiaauuii*eamd*rn*rcstr**uci criptam	Incercam sa lucram cu coduri si criptari.

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

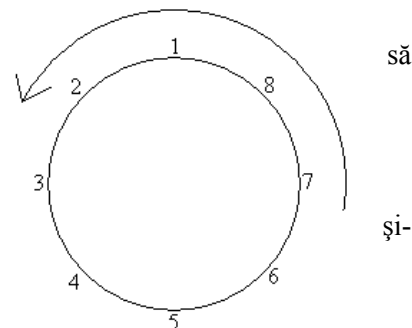
## Ziua 2

sursa: masina.pas, masina.cpp, masina.c

## mașina

## Clasa a IX-a

O țară are  $3 \leq N \leq 30\,000$  orașe, numerotate de la 1 la  $N$ , dispuse pe un cerc. PAM tocmai și-a luat carnet de conducere și vrea viziteze toate orașele țării. Lui PAM îi este frică să conducă prin locuri aglomerate așa că ea și-a propus să meargă numai pe șoselele unde traficul este mai redus. Există șosele de legătură între oricare două orașe alăturate: între orașul 1 și orașul 2, ..., între orașul  $i$  și orașul  $i+1$ , iar orașul  $N$  este legat de orașul 1. Ca să nu se rătăcească, PAM a propus să-și aleagă un oraș de început și să meargă pe șoselele respective în sens trigonometric până ajunge înapoi în orașul de unde a plecat. Dacă PAM pleacă din orașul  $K$ , atunci traseul ei va fi:  $K, K+1, \dots, N, 1, 2, \dots, K$ .



Mașina lui PAM are un rezervor foarte mare (în care poate pune oricât de multă benzină).

În fiecare oraș, PAM ia toată cantitatea de benzină existentă în oraș, iar parcurgerea fiecărei șosele necesită o anumită cantitate de benzină.

## Cerință

Știind că PAM are, la începutul călătoriei, doar benzina existentă în orașul de plecare, și că, atunci când ajunge într-un oraș, ea va lua toată cantitatea de benzină disponibilă în acel oraș, să se găsească un oraș din care PAM își poate începe excursia astfel încât să nu rămână fără benzină.

Se consideră că PAM a rămas fără benzină dacă în momentul plecării dintr-un oraș, nu are suficientă benzină pentru a parcurge șoseaua care duce la orașul următor. Dacă benzina îi ajunge la fix (adică la plecare are tot atâta benzină câtă îi trebuie) se consideră că PAM poate ajunge până în orașul următor.

## Date de intrare

Fișierul de intrare **masina.in** conține pe prima linie numărul  $N$ . Pe cea de-a doua linie se găsesc  $N$  numere naturale  $a[1], a[2], \dots, a[N]$ , separate prin câte un spațiu, unde  $a[i]$  reprezintă cantitatea de benzină disponibilă în orașul  $i$ . Linia a treia conține un șir de  $N$  numere naturale  $b[1], b[2], \dots, b[N]$ , separate prin câte un spațiu, unde  $b[i]$  reprezintă cantitatea de benzină necesară străbaterii șoselei dintre orașele  $i$  și  $i+1$  (sau  $N$  și 1, dacă  $i=N$ ).

## Date de ieșire

Fișierul de ieșire **masina.out** va conține un singur număr  $s$  care reprezintă un oraș din care, dacă PAM își începe călătoria, poate completa turul țării fără a face pana prostului.

## Observații

- Dacă există mai multe soluții, se cere una singură.
- $0 \leq a[i] \leq 30000$
- $1 \leq b[i] \leq 30000$
- $\sum_{i=1}^n a[i] \leq 2\,000\,000\,000$

## Exemplu

<b>masina.in</b>	<b>masina.out</b>
6 0 3 2 5 10 5 7 8 3 2 1 4	4

**Timp maxim de execuție:** 0.3 sec/test

## Ziua 2

sursa: operatii.pas, operatii.cpp, operatii.c

## operații

## Clasa a IX-a

Notăm cu  $c$  și  $r$  câtul și respectiv restul împărțirii unui număr  $nr$  la  $2^k$ , unde  $k$  este un număr natural nenul. Asupra numărului putem efectua succesiv următoarele operații:

$O1(nr, k)$  reprezintă transformarea numărului  $nr$  în numărul  $2^k(2c+1)+r$  pentru orice rest  $r$   
sau

$O2(nr, k)$  reprezintă transformarea numărului  $nr$  în numărul  $2^{k-1}c+r$  doar dacă  $r < 2^{k-1}$

## Cerință

Se dau  $m$  și  $n$  două numere naturale nenule.

Efectuați asupra numerelor  $m$  și  $n$  operații succesive,  $O1$  sau  $O2$ , pentru valori alese ale lui  $k$ , astfel încât după un număr finit de operații cele două numere să devină egale, iar valoarea astfel obținută să fie minimă.

## Date de intrare

Fișierul de intrare **operatii.out** conține pe o singură linie:

$m$   $n$  două numere naturale nenule, separate printr-un spațiu, reprezentând cele două numere date.

## Date de ieșire

Fișierul de ieșire **operatii.out** conține pe cele  $i+j+3$  linii următoarele:

	Numărul natural nenul <b><i>nmin</i></b> , reprezentând valoarea minimă obținută din $m$ și $n$ prin
$nmin$	aplicarea unor succesiuni de operații $O1$ sau $O2$ , pe prima linie
$i$	Numărul operațiilor efectuate asupra primului număr $m$ , pe a doua linie
$op_1$ $k_1$	Pe următoarele $i$ linii:
...	perechi de numere reprezentând operația (1 sau 2) și respectiv valoarea lui $k$ pentru operația
$op_i$ $k_i$	respectivă, separate printr-un spațiu.
$j$	Numărul operațiilor efectuate asupra celui de al doilea număr $n$ , pe linia $i+2$
$op_1$ $k_1$	Pe următoarele $j$ linii:
...	perechi de numere reprezentând operația (1 sau 2) și respectiv valoarea lui $k$ pentru operația
$op_j$ $k_j$	respectivă, separate printr-un spațiu

## Restricții

- $1 < m, n \leq 2\,000\,000\,000$  (două miliarde)

## Exemplu

**OPERATII.IN**

11 45

**OPERATII.OUT**

15

2

2 3

1 2

2

2 2

2 4

Timp maxim de executare/test: 1 secundă