



Concursul INTERJUDEȚEAN de programare GRIGORE MOISIL

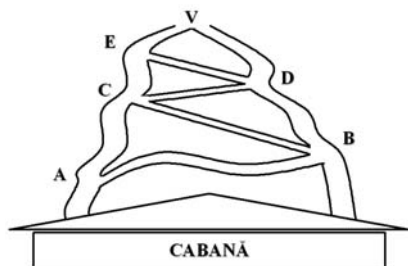
Ediția din acest an a concursului interjudețean de programare Grigore Moisil a avut loc la Satu Mare pe data de 29 martie 2003. În continuare vă prezentăm enunțurile celor șapte probleme propuse spre rezolvare la acest concurs.

Clasa a IX-a

P060319: Excursie

Pe un vârf înalt din Munții Retezat se poate urca pe două poteci abrupte care pornesc de la cabană. Cele două poteci sunt legate din când în când printr-o serpentină, adică o potecă mai comodă care leagă două puncte oarecare de pe cele două poteci abrupte, astfel încât pe aceste porțiuni se poate urca mai domol.

Un grup de turiști ar vrea să știe în câte feluri ar putea să urce spre vârf, folosind toate modalitățile posibile de a urca pe porțiuni de potecă abruptă, apoi pe porțiuni de serpentine, iar pe potecă abruptă etc.



Să se determine numărul tuturor traseelor pe care turiștii ar putea să urce spre vârf, știind că ei vor merge întotdeauna în sus spre vârf. De exemplu, corespunzător figurii sunt posibile porțiunile $cabană \rightarrow A$, $A \rightarrow C$, sau $A \rightarrow B$, apoi $cabană \rightarrow B$ etc., dar nu sunt posibile $C \rightarrow A$, $B \rightarrow A$ etc.

Date de intrare

În fișierul de intrare **EXCURSIE.IN** se află un număr natural n , reprezentând numărul total de intersecții ale potecilor abrupte și ale serpentinei.

Date de ieșire

În fișierul de ieșire **EXCURSIE.OUT** se va scrie un singur număr, reprezentând numărul traseelor diferite dintre punctul de pornire și vârf.

Restricție

- $1 \leq n \leq 40$.

Exemplu

EXCURSIE.IN

5

EXCURSIE.OUT

13

Timp de execuție: 1 secundă/test

P060320: Loto

Vasile ar vrea cu orice preț să câștige la *loto*. El își notează toate numerele posibile de jucat, apoi scrie un program care să-i genereze configurații de numere care ar putea să fie trecute pe biletele de *loto*. Programul lui generează toate variantele posibile (combinații de 6 numere din 49) și de asemenea, numerotează variantele începând cu numărul de ordine 1. Vasile și-a dat seama că numărul variantelor fiind foarte mare, nu are nici o șansă să cumpere atâtea bilete câte variante sunt. În concluzie, el hotărăște să aleagă prima variantă, o trece pe un bilet, apoi selectează fiecare a k -a variantă din lista generată de program.

Atunci când ajunge la capătul listei, continuă numărarea de la începutul listei, ca și cum aceasta ar fi circulară. El termină selectarea variantelor atunci când, datorită nu-



mărătorii efectuate, nimerește o variantă pe care a ales-o deja.

Vasile totuși este supărat și își imaginează un joc de *loto* în care se extrag cel mult y numere din totalul de x .

Ajutați-l pe Vasile și scrieți un program care calculează numărul total n al variantelor pe baza numerelor x și y .

Scrieți un alt program, care pe baza numerelor n și k date, determină numărul variantelor pe care Vasile nu le va juca dintre cele posibile, dacă el va aplica strategia descrisă.

Date de intrare

Pe prima linie a fișierului de intrare **LOTO1.IN** sunt scrise numerele naturale nenule x și y , despărțite printr-un spațiu.

Pe prima linie a fișierului de intrare **LOTO2.IN** se află două numere naturale nenule n și k , despărțite printr-un spațiu, reprezentând numărul de variante și numărul k ales de Vasile.

Date de ieșire

În fișierului de ieșire **LOTO1.OUT** veți scrie numărul variantelor calculat pe baza numerelor x și y , citite din fișierul de intrare.

În fișierul de ieșire **LOTO2.OUT** veți scrie numărul variantelor pe care nu le va juca Vasile.

Restricții și precizări

- două variante care conțin aceleași numere, dar în ordine diferită, se consideră identice;
- $1 < n \leq 13983816$;
- $1 < k < n$;
- $1 < x \leq 49$;
- $1 < y \leq 6$.

Exemple

LOTO1.IN	LOTO2.IN
4 2	6 2
LOTO1.OUT	LOTO2.OUT
6	3
LOTO1.IN	LOTO2.IN
30 4	27405 5
LOTO1.OUT	LOTO2.OUT
27405	21924

Timp de execuție: 1 secundă/test

Clasa a X-a

P060321: Împărțeală

Ancuța a primit de ziua ei n bomboane și ar vrea să le împartă printre prietenii cei mai apropiați. În scopul unei împărțeli speciale, ea a notat pe o hârtie numele celor k prieteni în ordinea importanței lor în viața ei și ar vrea ca:

- fiecare prieten să primească cel puțin o bomboană, și

- un prieten nu poate primi mai multe bomboane decât un altul care are numărul de ordine mai mic.

Observând cât de multe posibilități are, *Ancuța* s-a zăpăcit, dar totuși ar vrea să știe din câte posibilități va trebui să aleagă. Ajutați-o, scriind un program care calculează numărul posibilităților de a împărți cele n bomboane printre cei k prieteni, respectând cele două restricții stabilite de *Ancuța*.

Date de intrare

Fișierul de intrare **ANCA.IN** conține două numere naturale, despărțite printr-un spațiu, reprezentând numărul bomboanelor și numărul prietenilor.

Date de ieșire

În fișierul de ieșire **ANCA.OUT** se va scrie un număr natural care reprezintă numărul posibilităților pe care le are *Ancuța* la dispoziție pentru a împărți bomboanele printre prietenii săi.

Restricții

- $1 \leq n \leq 100$;
- $1 \leq k \leq 12$;
- trebuie împărțite toate bomboanele.

Exemplu

ANCA.IN
10 3

ANCA.OUT
8

Timp de execuție: 1 secundă/test

P060322: Cutii

S-a mutat muzeul. Obiectele au fost împachetate în cutii având forme cubice de diverse dimensiuni. La despachetare lucrează multe persoane în paralel și pentru a evita dezordinea, prin încăperile unde se lucrează la despachetare s-a instalat o bandă rulantă pe care se așează cutiile goale, cu singura deschizătură orientată în sus.

Ștefan stă la capătul benzii și strânge cutiile. El a primit sarcina să împacheteze cutiile unele în altele, astfel încât numărul pachetelor de cutii să fie cel mai mic posibil.

Directorul muzeului, văzându-l pe *Ștefan* dezorientat datorită acestei cerințe, s-a hotărât să-l ajute și a stabilit regulile următoare:

- cutiile se culeg de pe bandă în ordinea sosirii lor;
- cutia curentă se așează într-o altă cutie, dacă aceasta are dimensiunea mai mică;
- dacă nu există pachet început în care să încapă cutia curentă, aceasta va constitui prima cutie dintr-un pachet nou;
- într-un pachet început se așează o singură cutie (nu se pun mai multe cutii unele lângă altele, chiar dacă există loc);
- o cutie așezată la un moment dat nu se mai scoate;



- un pachet început nu se așează în alt pachet, chiar dacă acest lucru ar fi posibil;
- nici o cutie nu poate fi ignorată.

Scrieți un program care determină numărul minim de pachete de cutii rezultate în urma muncii lui Ștefan, precum și secvențele de cutii din fiecare pachet.

Date de intrare

Prima linie a fișierului de intrare **CUTII.IN** conține un singur număr natural n , reprezentând numărul cutiilor. Următoarele n linii conțin fiecare câte un număr natural, reprezentând dimensiunile cutiilor.

Date de ieșire

Pe prima linie a fișierului **CUTII.OUT** se va scrie numărul minim m de pachete de cutii. Pe următoarele m linii se vor scrie dimensiunile cutiilor pe care Ștefan le va împacheta într-un singur pachet, în ordinea împachetării.

Restricții

- $0 \leq n \leq 1000$;
- $1 \leq \text{dimensiune_cutie} \leq 10.000$.

Exemplu

CUTII.IN

```
10
4
1
5
10
7
9
2
8
3
2
```

CUTII.OUT

```
4
4 1
5 2
10 7 3 2
9 8
```

Timp de execuție: 1 secundă/test

Clasele a XI-a și a XII-a

P060323: Ali Baba și cei 40 de hoți

Ali Baba și-a ascuns mărgelile adunate în boluri într-un loc numai de el știut. Din păcate un hoț l-a pândit și i-a condus pe prieteni la locul respectiv. Hoții, având un strop de conștiință, au decis să nu fure toate mărgelile ci doar o parte mică, în așa fel încât Ali Baba să nu observe nimic. Hoții au hotărât să confecționeze un șirag din acele tipuri

de mărgelile din care există cele mai multe. Au decis să verifice toate bolurile și să aleagă câte k mărgelile identice (având aceeași culoare) din fiecare bol, dacă mai mult decât jumătate din mărgelile au aceeași culoare și dacă există cel puțin k mărgelile de aceeași culoare în bolul respectiv.

Hoții au lucrat la numărătoare, unii au reușit să pună deoparte mărgelile, alții au ajuns la concluzia că nu pot lua mărgelile din bolul pe care l-au verificat. În final, văzând că au reușit să adune destul de multe mărgelile, unul dintre ei a propus să confecționeze șiragul doar din mărgelile având aceeași culoare, dar și de data aceasta numai în situația în care, dintre mărgelile alese, mai mult decât jumătate au aceeași culoare.

Cunoscând numărul bolurilor și culorile mărgelilor din fiecare bol, stabiliți dacă hoții au putut sau nu să confecționeze șiragul de mărgelile conform regulilor stabilite.

Date de intrare

Pe prima linie a fișierului de intrare **MARGELE.IN** s-a scris numărul n de boluri și numărul k reprezentând numărul de mărgelile luate din fiecare bol în scopul confecționării șiragului de mărgelile. Pe următoarele n linii sunt scrise culorile mărgelilor, despărțite printr-un spațiu (pe linia i sunt culorile mărgelilor din bolul i).

Date de ieșire

În cazul în care hoții vor reuși să confecționeze șiragul, pe prima linie a fișierului de ieșire **MARGELE.OUT** veți scrie culoarea șiragului, în caz contrar veți scrie NU. În cazul în care în fișier nu s-a scris NU, pe cea de a doua linie veți scrie numărul mărgelilor din șirag.

Restricții și precizări

- $1 < k \leq 500$;
- numărul de boluri este cel mult 40;
- într-un bol toate mărgelile pot fi de culori diferite;
- într-un bol pot fi cel mult 2000 de mărgelile;
- culoarea este specificată printr-o succesiune de cel mult 10 litere ale alfabetului latin.

Exemplu

MARGELE.IN

```
5 3
rosu rosu verde rosu
verde galben galben galben
albastru rosu alb
rosu rosu rosu albastru rosu
rosu albastru rosu
```

MARGELE.OUT

```
rosu
6
```

Explicație

Din primul bol se pot lua 3 mărgelile roșii, din al doilea bol se pot lua 3 mărgelile galbene, din cel de-al treilea bol nu se



pot lua 3 mărgelile de culoare identică, din bolul 4 se vor lua 3 mărgelile roșii, din bolul 5 nu se pot lua 3 mărgelile de culoare identică. În total sunt 6 mărgelile roșii și 3 galbene, deci șiragul va fi confecționat din 6 mărgelile roșii.

Timp de execuție: 1 secundă/test

P060324: Profi și studenți

În universități, fiecare cadru didactic (cu excepția rectorului) are un singur șef direct și poate avea subalterni. Se știe că dacă rectorul nu este șeful direct al unui cadru didactic, atunci el este sigur șeful indirect al acestuia.

La sfârșitul ședințelor profesorii universitari mai zăbovesc un pic în sala de conferințe, apoi părăsesc sala unul câte unul. Studenții au observat că profesorii părăsesc sala pe baza unor reguli:

- nici o persoană nu părăsește sala atâta timp cât în sală mai există persoane care le sunt subordonate;
- dacă la un moment dat ar fi posibil să iasă din sală mai multe persoane, atunci va ieși cea care are numele (numărul de ordine în tabelul nominal) mai mic;
- fiecare persoană își salută șeful direct în momentul în care părăsește sala.

La mai multe universități studenții au numerotat profesorii pe baza ordinii alfabete și au notat numerele de ordine ale cadrelor didactice salutate (este posibil ca un același număr să fie notat de mai multe ori) și vor să reconstituie ordinea în care profesorii au părăsit sala.

Date de intrare

Pe prima linie a fișierului **PROFI.IN** este scris numărul m al universităților. Următoarele $2 \cdot m$ linii conțin datele referitoare la cele m universități. Pe prima linie din a i -a pereche se află scris numărul n al cadrelor didactice din a i -a universitate. Pe cea de a doua linie din fiecare pereche de linii sunt scrise numerele de ordine ale profesorilor în ordinea în care au fost salutați.

Date de ieșire

Fișierul de ieșire **PROFI.OUT** va conține m linii. Răspunsul referitor la o universitate va conține un șir de numere naturale, despărțite printr-un spațiu, reprezentând numerele de ordine ale cadrelor didactice în ordinea în care au părăsit sala.

Restricții și precizări

- $1 \leq m \leq 100$;
- $2 \leq n \leq 10.000$, pentru orice universitate.

Exemplu

PROFI.IN

```
3
5
2 1 4 2
6
```

```
1 1 1 1 1
```

```
5
```

```
2 2 5 2
```

PROFI.OUT

```
3 5 1 4 2
```

```
2 3 4 5 6 1
```

```
1 3 4 5 2
```

Timp de execuție: 1 secundă/test

P060325: Hacker

Un *hacker* a găsit o matrice de caractere care este folosită de *FBI* la criptarea unor informații secrete. Pentru a putea decifra o informație importantă, el are nevoie de o cheie, care este un șir de caractere și de numărul de apariții al acestuia în matrice. Un șir de caractere apare în matrice dacă pozițiile sale se parcurg începând de la o poziție (i, j) fie la dreapta, fie la stânga, respectiv fie în sus, fie în jos, cât timp este nevoie pentru a regăsi fiecare caracter a cheii în ordinea elementelor aflate în matrice.

Membrii comisiei au aflat cheile de criptare și lucrează cu *hackerul* pentru spargerea datelor *FBI*-ului. Ajutați comisia să rezolve problema, determinând numărul de apariții a cheii.

Date de intrare

Pe prima linie a fișierului **HACKER.IN** este scris numărul n , reprezentând dimensiunea matricei. Următoarea linie conține șirul de caractere care reprezintă cheia de criptare. Următoarele n linii conțin câte n caractere, corespunzătoare matricei *FBI*-ului.

Date de ieșire

Fișierul de ieșire **HACKER.OUT** va conține pe prima linie numărul de apariții ale cheii în matricea dată la intrare.

Restricții și precizări

- $2 \leq n \leq 100$;
- în cheie caracterele sunt distincte;
- caracterele în cheie și în matrice sunt litere mari și mici ale alfabetului englezesc.

Exemplu

HACKER.IN

```
4
BCD
AABA
ABCD
AADA
AAAA
```

HACKER.OUT

```
4
```

Timp de execuție: 1 secundă/test