



Olimpiada JUDEȚEANĂ de INFORMATICĂ 2005

Etapa județeană a Olimpiadei de Informatică din acest an s-a desfășurat în ultima săptămână a lunii februarie. În cele ce urmează vom prezenta enunțurile problemelor propuse cu care s-au confruntat participanții de la acest concurs.

Clasa a V-a

P050201: Ultima cifră

Doru Popescu Anastasiu

Fie n un număr natural și suma s definită astfel:

$$s = 1 + 2^2 + 3^3 + \dots + n^n.$$

Scrieți un program care să afișeze ultima cifră a valorii s .

Date de intrare

De la tastatură se citește valoarea n .

Date de ieșire

La ieșirea standard se va afișa numai ultima cifră a lui s .

Restricții și precizări

Numărul n este întreg și este cuprins între 0 și 101 (exclusiv).

Exemplu

Intrarea standard (tastatură)

$n=3$

Ieșirea standard (ecran)

2

Explicație

Suma este 32, deci ultima cifră este 2.

Timp de execuție/test: 1 secundă

P050202: Mulțimi

Dana Vlădoiu

Se consideră n mulțimi; fiecare dintre aceste mulțimi conține numai numere consecutive.

Pentru a descrie aceste mulțimi este suficient să precizăm primul și ultimul element.

Scrieți un program care să determine elementele intersecției celor n mulțimi.

Date de intrare

De la tastatură se citește numărul n și perechi de numere, câte una pe un rând, separate prin câte un spațiu, care reprezintă cel mai mic, respectiv cel mai mare element din fiecare mulțime.

Date de ieșire

Pe ecran se vor afișa pe prima linie elementele intersecției mulțimilor precizate, separate prin câte un spațiu.

Restricții și precizări

Numărul n al mulțimilor este cuprins între 0 și 31 (exclusiv).

Toate elementele acestor mulțimi sunt numere naturale mai mici decât 41.

Exemplu

Intrarea standard (tastatură)

$n=3$

5 10

4 11

2 9

Ieșirea standard (ecran)

5 6 7 8 9

Explicație

Avem mulțimile

$\{5, 6, 7, 8, 9, 10\}$

$\{4, 5, 6, 7, 8, 9, 10, 11\}$

$\{2, 3, 4, 5, 6, 7, 8, 9\}$

deci, intersecția este:

$\{5, 6, 7, 8, 9\}$

Timp de execuție/test: 1 secundă

Clasa a VI-a

P050203: Numere

Roxana Țimblaru

Se consideră două numere naturale a și b , precum și reprezentările binare ale acestora.

Reprezentărilor binare li se aplică următoarea transformare: dacă prima cifră (cea mai din stânga) din reprezentarea binară a numărului a este egală cu ultima cifră (cea mai din dreapta) din reprezentarea binară a



numărului b , atunci se elimină prima cifră (cea mai din stânga) din reprezentarea binară a numărului a și ultima cifră (cea mai din dreapta) din reprezentarea binară a numărului b și se continuă transformările în același mod până când prima cifră (cea mai din stânga) din reprezentarea binară a numărului a este diferită de ultima cifră (cea mai din dreapta) din reprezentarea binară a numărului b .

Valorile rămase după transformările suferite sunt convertite în baza 10 și se obțin două numere c și d .

Scrieți un program care citește numerele a și b și care afișează valoarea obținută prin însumarea celor două numere c și d .

Date de intrare

Cele două valori a și b se citesc de la tastatură.

Date de ieșire

Rezultatul va fi un număr natural care se va afișa pe ecran.

Restricții și precizări

Numerele a și b sunt naturale și sunt cuprinse între valorile 0 și 30000 (exclusiv).

Dacă asupra celor două reprezentări binare nu s-a efectuat nici o transformare, întrucât prima cifră din reprezentarea numărului a este diferită de ultima cifră din reprezentarea binară a numărului b , atunci numărul c va fi identic cu numărul a , iar d cu numărul b .

Dacă în urma unei transformări se elimină și ultima cifră din reprezentarea binară, atunci rezultatul final va fi 0.

Exemplu

Intrarea standard (tastatură)

$a=13$

$b=27$

Ieșirea standard (ecran)

1

Intrarea standard (tastatură)

$a=13$

$b=25$

Ieșirea standard (ecran)

17

Intrarea standard (tastatură)

$a=13$

$b=20$

Ieșirea standard (ecran)

33

Explicații

În cazul primului exemplu, numărul 13 se scrie în baza 2 sub forma 1101, iar numărul 27 sub forma 11011.

După prima transformare se obțin: 101, respectiv 1101.

Se continuă transformările și se obțin: 01, respectiv 110.

Se continuă transformarea și se obține 1, respectiv 11. Se continuă transformarea și se obțin: 0 și 1.

Se face conversia în baza 10 și se obțin: $c = 0$ și $d = 1$. Deci, suma $c + d$ este 1.

În cazul celui de-al doilea exemplu, numărul 13 se scrie în baza 2 sub forma 1101, iar numărul 25 sub forma 11001.

După prima transformare se obțin: 101, respectiv 1100.

Din acest moment nu se mai pot face transformări. Se reprezintă 101 în baza 10 și se obține 5, iar 1100 se reprezintă în baza 10 și se obține 12.

Deci $c = 5$ și $d = 12$, iar suma $c + d$ este 17.

În cazul celui de-al treilea exemplu, numărul 13 se scrie în baza 2 sub forma 1101, iar numărul 20 sub forma 10100.

Se observă că prima cifră din reprezentarea lui a este diferită de ultima cifră din reprezentarea lui b .

Deci, nu se face nici o transformare. Se obțin după conversia în baza 10: $c = 13$ și $d = 20$. Deci, suma $c + d$ este 33.

Timp de execuție/test: 1 secundă

P050204: Șir

Dan Grigoriu

Se consideră numărul natural k . Dorim să obținem un tablou unidimen-

sional a , cu elemente naturale obținute astfel: a_1 este un număr format din două cifre (cifra zecilor lui a_1 este cifra sutelor produsului $k \cdot k$, iar cifra unităților lui a_1 este cifra zecilor produsului $k \cdot k$).

Pentru $i > 1$, a_i se obține astfel: a_i este un număr format din două cifre (cifra zecilor lui a_i este cifra sutelor produsului $a_{i-1} \cdot a_{i-1}$, iar cifra unităților a lui a_i este cifra zecilor produsului $a_{i-1} \cdot a_{i-1}$).

Procesul de generare a termenilor tabloului se încheie în momentul când este generat un număr care a mai fost generat înainte.

Ultimul număr (cel care se repetă) nu face parte din tablou.

Este posibil ca numerele numite în text ca fiind formate din două cifre să aibă de fapt o singură cifră, în cazul în care cifra zecilor lor este 0; ele pot fi chiar și 0.

Scrieți un program care:

- să determine tabloul a ;
- să determine elementele tabloului obținut, sortate crescător după prima lor cifră (cea mai din stânga).

Date de intrare

De la tastatură se citește numărul natural k .

Date de ieșire

Pe prima linie a ecranului se vor afișa elementele tabloului a , în ordinea generării lor, separate de un spațiu.

Pe a doua linie a ecranului se vor afișa elementele tabloului a , în ordinea cerută la cerința a doua; elementele tabloului vor fi separate de câte un spațiu.

Restricții și precizări

Valoarea k este un număr natural cuprins între 11 și 999.

În cazul celei de-a doua cerințe, dacă două sau mai multe elemente din tabloul a au aceeași primă cifră, atunci aceste elemente se pot afișa în orice ordine care respectă restricțiile impuse.

În exemplul de mai jos, afișarea pentru cerința a doua putea apărea și sub forma: 0 2 25 5 62 84, deci

putem interschimba valorile 2 și 25, deoarece ambele au prima cifră 2; în acest caz nu mai există alte posibilități de afișare.

Exemplu

Intrarea standard (tastatură)

k=16

Ieșirea standard (ecran)

25 62 84 5 2 0

0 25 2 5 62 84

Explicații

Determinarea tabloului a se face astfel:

$k \cdot k = 16 \cdot 16 = 256$, deci $a_1 = 25$;
 $25 \cdot 25 = 625$, deci $a_2 = 62$;
 $62 \cdot 62 = 3844$, deci $a_3 = 84$
 $84 \cdot 84 = 7056$, deci $a_4 = 5$;
 $5 \cdot 5 = 25$, deci $a_5 = 2$;
 $2 \cdot 2 = 4$, deci $a_6 = 0$
 $0 \cdot 0 = 0$.

Aici se oprește generarea tabloului cu șase elemente, care se afișează.

În continuare, prima cifră a lui a_1 este 2, a lui a_2 este 6, a lui a_3 este 8, a lui a_4 este 5, a lui a_5 este 2 și a lui a_6 este 0.

În urma sortării acestor prime cifre: 2 (asociată cu a_1), 6 (asociată cu a_2), 8 (asociată cu a_3), 5 (asociată cu a_4), 2 (asociată cu a_5) și 0 (asociată cu a_6), se obține ordinea nouă a acestor numere: 0, 2, 2, 5, 6, 8, asociate respectiv cu elementele din tabloul a : 0, 25, 2, 5, 62, 84; elementele din a se vor afișa în această ordine, sau în ordinea: 0, 2, 25, 5, 62, 84.

Timp de execuție/test: 1 secundă

Clasa a VII-a

P050205: OCR

Marinel Șerban

O imagine va fi reprezentată ca un tablou dreptunghiular de numere reale, fiecare număr reprezentând o valoare pe scala de gri a imaginii. Valorile sunt cuprinse între 0 (corespunzând unei regiuni total albe) și 1 (pentru zona total neagră), cu două zecimale.

Centrul de greutate al imaginii este un element al tabloului. Să presupunem că acesta se află pe linia i și coloana j .

În acest caz, valoarea absolută a diferenței dintre suma elementelor din zona aflată deasupra liniei i și suma elementelor din zona aflată sub linia i , este minimă.

În mod analog, valoarea absolută a diferenței dintre suma elementelor din stânga coloanei j și suma elementelor din dreapta coloanei j trebuie să fie de asemenea minimă.

Să considerăm ca exemplu următorul tablou care poate proveni din scanarea literei 'o'.

Centrul de greutate este pe linia 3 și coloana 3, deoarece diferența sumelor elementelor din fiecare zonă formată ignorând linia a treia este 0.1 (sumele sunt 5.55 și 5.65) și de asemenea, diferența sumelor elementelor fiecărei zone formate ignorând coloana a treia este 0.1 (sumele sunt 5.60 și 5.70).

0.7	0.75	0.7	0.75	0.8
0.55	0.3	0.2	0.1	0.7
0.8	0.1	0.1	0.1	0.8
0.7	0.0	0.0	0.0	0.8
0.8	0.9	0.8	0.75	0.9

Scrieți un program care să determine centrul de greutate al unei imagini scanate.

Date de intrare

Fișierul de intrare **ocr.in** conține reprezentarea unei imagini.

Prima linie a fișierului de intrare conține două numere naturale n și m , separate printr-un spațiu, care reprezintă numărul de linii, respectiv numărul de coloane ale tabloului care reprezintă imaginea.

Următoarele n linii conțin câte m numere reale din intervalul $[0, 1]$, separate prin câte un spațiu, care reprezintă elementele tabloului.

Date de ieșire

Fișierul de ieșire **ocr.out** va conține o singură linie pe care se găsesc două numere naturale l și c , separate printr-un spațiu, reprezentând coordona-

tele (linie, coloană) centrului de greutate.

În cazul în care sunt determinate mai multe centre de greutate, se vor afișa coordonatele celui cu indicele de linie maxim; dacă există mai multe centre de greutate pe aceeași linie, se va afișa cel cu indicele de coloană maxim.

Restricții și precizări

Numerele n și m sunt naturale cuprinse între 1 și 50.

Valorile reale sunt exprimate cu maximum două zecimale.

Liniile sunt numerotate de la 1 la n (de sus în jos), iar coloanele de la 1 la m (de la stânga la dreapta).

Exemplu

ocr.in

```
5 5
0.1 0.2 0.1 0.2 0.1
0.1 0.2 0.3 0.1 0.1
0.2 0.3 0.1 0.1 0.3
0.4 0.1 0.1 0.1 0.2
0.2 0.2 0.3 0.3 0.1
```

ocr.out

```
3 3
```

ocr.in

```
5 10
0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3
0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.6
```

ocr.out

```
4 6
```

Timp de execuție/test: 1 secundă

P050206: Tabel

Daniel Popa

După cum probabil știți, contabilii își țin datele sub forma unor tabele și calculează tot felul de sume pe linii și pe coloane.

Contabilul nostru *Atnoc* și-a organizat valorile sub forma unui tabel cu n linii (numerotate de la 1 la n) și m coloane (numerotate de la 1 la m).

Elementele de pe ultima coloană sunt sumele elementelor de pe linii





(mai exact, elementul de pe linia i și coloana m este egal cu suma elementelor de pe linia i aflate pe coloanele 1, 2, ..., $m - 1$), iar elementele de pe ultima linie sunt sumele elementelor de pe coloane (mai exact, elementul de pe linia n și coloana i este egal cu suma elementelor de pe coloana i aflate pe liniile 1, 2, ..., $n - 1$).

Un exemplu de astfel de tabel este următorul:

```

2   5   7  14
11  6   6  23
13 11  13  37

```

Din păcate, *Atnoc* a stropit cu apă minunatul său tabel și astfel o parte dintre numerele din tabel au devenit ilizibile.

Scrieți un program care să reconstituie toate datele din tabel.

Date de intrare

Pe prima linie a fișierului text de intrare **tabel.in** se află două numere naturale n și m , separate printr-un spațiu, care reprezintă numărul de linii și respectiv numărul de coloane ale tabelului.

Pe cea de a doua linie a fișierului de intrare se află un număr natural p care reprezintă numărul de valori nedeteriorate din tabel.

Pe fiecare dintre următoarele p linii se află câte trei numere naturale, separate prin câte un spațiu l , c și v , unde l este numărul liniei, c este numărul coloanei și v este valoarea elementului de pe linia l și coloana c din tabel.

Date de ieșire

În fișierul text de ieșire **tabel.out** se va scrie tabelul reconstituit, pe n linii câte m valori separate prin câte un spațiu.

Restricții și precizări

Numerele n și m sunt naturale cuprinse între 2 și 50.

Valorile din tabel sunt numere naturale mai mici decât 32000.

Datele din tabel pot fi reconstituite întotdeauna.

Exemplu

tabel.in

```

3 4
10
1 1 2
1 2 5
1 3 7
1 4 14
2 2 6
2 4 23
3 1 13
3 2 11
3 3 13
3 4 37

```

tabel.out

```

2 5 7 14
11 6 6 23
13 11 13 37

```

Timp de execuție/test: 1 secundă

Clasa a VIII-a

P050207: Muzică

Maria Niță și Adrian Niță

Fiind elev la un liceu de artă, secția muzică, *Andrei* își propune să studieze o gamă nouă formată din 10 note muzicale.

Pasionat și de matematică el își propune pornind de la două numere naturale a și b ($a < b$) să compună o *Simfonie interminabilă*, generând un șir de note în gama cea nouă.

Astfel, el generează fiecare notă a simfoniei înmulțind pe a cu 10 și împărțind rezultatul la b (împărțire întreagă).

Pentru a nu genera aceeași notă, el modifică de fiecare dată pe a , înlocuindu-l cu restul împărțirii lui $a \cdot 10$ la b .

Așadar, notele sunt generate după regula $a \cdot 10 \text{ div } b$, unde după fiecare pas, a se schimbă astfel: $a = a \cdot 10 \text{ mod } b$ (operația **div** reprezintă câtul întreg al împărțirii, iar operația **mod** este întregul care reprezintă restul împărțirii).

Astfel, pornind de la $a = 42$ și $b = 130$, el va genera notele: 3 2 3 0 7 6 9 2 3 0 7 6 9 2 etc.

```

3 = 42 · 10 div 130;
a = 42 · 10 mod 130, a = 30

```

```

2 = 30 · 10 div 130;
a = 300 mod 130, a = 40
3 = 40 · 10 div 130;
a = 400 mod 130, a = 10
0 = 10 · 10 div 130;
a = 100 mod 130, a = 100
7 = 100 · 10 div 130;
a = 1000 mod 130, a = 90
6 = 90 · 10 div 130;
a = 900 mod 130, a = 120
9 = 120 · 10 div 130;
a = 1200 mod 130, a = 30
2 = 30 · 10 div 130;
a = 300 mod 130, a = 40
etc.

```

Ascultând simfonia, *Andrei* constată că, de la un moment dat, o secvență începe să se repete identic de un număr infinit de ori.

Andrei numește secvența formată de primele note, cele aflate înaintea secvenței care se repetă, *tema*, iar secvența care se repetă, *refrenul* simfoniei. De exemplu, în secvența anterioară, 3 este tema, iar 230769 este refrenul.

El consideră tema și refrenul cu lungimi cât mai mici posibil. Astfel, în exemplul anterior, nu se pot considera temă respectiv refren nici 32 și 307692, nici 3 și 230769230769.

Există și cazul în care nu există temă, adică simfonia începe direct cu refrenul.

Scrieți un program care, citind două numere naturale a și b ($a < b$), va determina cifrele temei și cifrele refrenului.

Se vor afișa cifrele temei, în continuare cifrele refrenului apoi un spațiu urmat de un număr reprezentând câte cifre are refrenul.

Date de intrare

Fișierul de intrare **muzica.in** conține pe prima linie două valori: a și b numere naturale, separate printr-un spațiu.

Date de ieșire

Fișierul de ieșire **muzica.out** va conține o singură linie cu cifrele temei urmate în continuare de cifrele refrenului și, după un spațiu, numărul de cifre ale refrenului.

Restricții și precizări

Numerele a și b sunt naturale cuprinse între 1 și 1000.

Numerele a și b sunt distincte.

Exemplu

muzica.in

164 824

muzica.out

19902912621359223300970873786
407766 34

muzica.in

13 32

muzica.out

406250 1

muzica.in

6 11

muzica.out

54 2

Explicație

În cazul primului exemplu, 1 reprezintă tema, iar șirul 99029126213... reprezintă refrenul.

În cazul celui de-al doilea exemplu, 40625 reprezintă tema, iar 0 reprezintă refrenul.

În cazul celui de-al treilea exemplu nu există temă, iar șirul 54 reprezintă refrenul.

Timp de execuție/test: 1 secundă

P050208: Volei

Rodica Pinte

Câțiva băieți, după un fotbal mic, au hotărât să participe la jocul liber de volei al fetelor.

Fetele sunt așezate în cerc și nu își schimbă locurile între ele.

Un băiat poate intra în jocul fetelor numai dacă se așează între două fete și este mai înalt (strict) decât amândouă.

Cunoscând numărul de fete și înălțimea fiecăreia, în ordinea în care se află ele pe cerc, numărul băieților și înălțimea fiecăruia, se cere să se determine un număr maxim de băieți care pot participa la joc și poziția ocupată de fiecare pe cerc.

Date de intrare

Fișierul de intrare **volei.in** conține pe prima linie numărul n , care reprezintă numărul fetelor care participă la joc.

Cea de-a doua linie conține n numere naturale nenule de cel mult 3 cifre, despărțite prin câte un spațiu, numere reprezentând înălțimile fetelor, în ordinea de pe cerc, în sensul acelor de ceasornic, pornind de la o față oarecare.

Cea de-a treia linie conține un număr natural m care reprezintă numărul de băieți care vor să intre în joc.

Cea de-a patra linie conține m numere naturale nenule care conțin cel mult trei cifre fiecare, despărțite prin câte un spațiu, numere reprezentând înălțimile băieților care vor să intre în joc.

Date de ieșire

Fișierul de ieșire **volei.out** va conține pe prima linie un număr natural k , care reprezintă numărul maxim de băieți care pot participa la joc.

Pe cea de-a doua linie $n + k$ numere naturale, despărțite prin câte un spațiu, numere reprezentând înălțimile jucătorilor de volei, în ordinea în care se află aceștia în cerc în sensul acelor de ceasornic, pornind de la aceeași față care apare prima în fișierul de intrare.

Înălțimile băieților sunt scrise între paranteze.

Restricții și precizări

Numărul fetelor și numărul băieților sunt numere întregi cuprinse între 1 și 2.000.

Dacă există mai multe posibilități de a insera k băieți în joc, se va scrie una singură.

Dacă numerele reprezentând înălțimile nu vi se par plauzibile, atunci puteți considera că ele reprezintă coeficientul de inteligență al fiecărei persoane.

Pentru afișarea valorii corecte a numărului k se acordă 40 de puncte, iar pentru soluția completă (k maxim și configurația corectă a celor $n + k$ înălțimi ale jucătorilor) se acordă 100 de puncte.

Exemple

volei.in

4
15 11 72 31
3
20 50 25

volei.out

2
15 (20) 11 72 31 (50)
sau
2
15 (25) 11 72 31 (50)

volei.in

6
15 8 20 19 16 30
3
20 30 16

volei.out

3
15 (16) 8 (30) 20 19 (20) 16 30
sau
3
15 (16) 8 20 (30) 19 (20) 16 30

volei.in

4
10 20 15 100
2
20 8

volei.out

0
10 20 15 100

Timp de execuție/test: 1 secundă

Clasa a IX-a

P050209: Numere

Doru Popescu Anastasiu

Mircea este pasionat de programare. El a început să rezolve probleme din ce în ce mai grele.

Astfel a ajuns la o problemă, care are ca date de intrare un tablou pătratic cu n linii și n coloane, componente tabloului fiind toate numerele naturale distincte cuprinse în intervalul $[1 \text{ și } n^2]$.

Pentru a verifica programul pe care l-a scris Mircea are nevoie de un fișier care să conțină tabloul respectiv.





După ce a creat acest fișier, fratele său, pus pe șotii îi umblă în fișier și îi schimbă câteva numere consecutive, cu numărul 0.

Când se întoarce de la joacă, *Mircea* constată cu stupeoare că programul nu mai funcționează pentru testul respectiv.

După câteva ore de depanare *Mircea* își dă seama că programul lui este corect și că fișierul de intrare are probleme.

Scrieți un program care să-l ajute pe *Mircea*, găsindu-i cel mai mic și cel mai mare dintre numerele consecutive schimbate de fratele său.

Date de intrare

În fișierul **numere.in** se dă pe prima linie n , iar pe următoarele n linii elementele tabloului, câte n elemente pe o linie, separate între ele prin câte un spațiu, după modificările făcute de fratele lui *Mircea*.

Date de ieșire

În fișierul **numere.out** se vor scrie pe un singur rând cu un singur spațiu între ele numerele cerute (primul fiind cel mai mic).

Restricții și precizări

Numărul n este cuprins în intervalul $[1, 500]$.

Fratele lui *Mircea* schimbă cel puțin un număr în fișier.

Numerele schimbate de fratele lui *Mircea* sunt mai mici sau cel mult egale cu 60000.

Exemplu

numere.in

```
3
5 0 7
0 0 1
6 9 8
```

party.out

```
2 4
```

Explicație

În fișierul de intrare au fost înlocuite cu 0 numerele 2, 3, 4.

Timp de execuție/test: 1 secundă

P050210: MaxD

Maria Niță și Adrian Niță

Fiind elev în clasa a IX-a, *George* își propune să studieze capitolul divizibilitate cât mai bine.

Ajungând la numărul de divizori asociat unui număr natural, constată că pot exista mai multe numere într-un interval dat, care au un același număr de divizori.

De exemplu, în intervalul $[1, 10]$, 6, 8 și 10 au același număr de divizori, egal cu 4. De asemenea, 4 și 9 au același număr de divizori, egal cu 3 etc.

Scrieți un program care pentru un interval dat determină care este cel mai mic număr din interval ce are număr maxim de divizori.

Dacă sunt mai multe numere cu această proprietate se cere să se determine numărul acestora.

Date de intrare

Fișierul de intrare **maxd.in** conține pe prima linie două numere a și b separate prin spațiu ($a \leq b$) reprezentând extremitățile intervalului în care este căutat numărul.

Date de ieșire

Fișierul de ieșire **maxd.out** va conține pe prima linie trei numere separate prin câte un spațiu min , $nrdiv$ și $contor$, unde min reprezintă cea mai mică valoare din interval care are număr maxim de divizori, $nrdiv$ reprezintă numărul de divizori ai valorii min , iar $contor$ reprezintă numărul de numere din intervalul citit care $nrdiv$ divizori.

Restricții și precizări

Numerele a și b sunt cuprinse între 1 și 1.000.000.000.

Diferența maximă dintre valorile a și b nu va depăși în nici o situație valoarea 10.000.

Exemple

maxd.in

```
2 10
```

maxd.out

```
6 4 3
```

maxd.in

```
200 200
```

maxd.out

```
200 12 1
```

Explicație

În cazul primului exemplu, 6 este cel mai mic număr din interval care are maxim de divizori egal cu 4 și sunt 3 astfel de numere 6, 8, 10.

În cazul celui de-al doilea exemplu, 200 are 12 divizori iar în intervalul $[200, 200]$ există un singur număr cu această proprietate.

Timp de execuție/test: 1 secundă

Clasa a X-a

P050211: Lăcusta

Nistor-Eugen Moț

Se consideră o matrice dreptunghiulară cu m linii și n coloane, cu valori naturale. Traversăm matricea pornind de la colțul din stânga-sus înspre colțul din dreapta-jos.

O traversare constă din mai multe deplasări. La fiecare deplasare se execută un salt pe orizontală și un pas pe verticală. Un salt înseamnă că putem trece de la o celulă la oricare alta aflată pe aceeași linie, iar un pas înseamnă că putem trece de la o celulă la celula aflată imediat sub ea. Excepție face ultima deplasare (cea în care ne aflăm pe ultima linie), când vom efectua doar un salt pentru a ajunge în colțul dreapta-jos, dar nu vom mai efectua și pasul corespunzător. Astfel, traversarea va consta din vizitarea a $2 \cdot m$ celule.

Scrieți un program care să determine suma minimă care poate fi obținută în urma unei astfel de traversări.

Date de intrare

Fișierul de intrare **lacusta.in** conține pe prima linie două numere naturale separate printr-un spațiu m și n , reprezentând numărul de linii și respectiv numărul de coloane ale matricei. Pe următoarele m linii este descrisă matricea, câte n numere pe fie-

care dintre aceste linii, separate prin câte un spațiu.

Date de ieșire

Fișierul de ieșire **lacusta.out** va conține o singură linie pe care va fi scrisă suma minimă găsită.

Restricții și precizări

Numeralele m și n sunt cuprinse între 1 și 100. Valorile elementelor matricii sunt numere întregi cuprinse în intervalul $[1, 255]$.

Exemplu

lacusta.in

```
4 5
3 4 5 7 9
6 6 3 4 4
6 3 3 9 6
6 5 3 8 2
```

lacusta.out

28

Explicație

Drumul este:

```
(1, 1) → (1, 3) → (2, 3) →
(2, 2) → (3, 2) → (3, 3) →
(4, 3) → (4, 5)
```

Timp de execuție/test: 1 secundă

P050212: Scara

Marinel Șerban și Emanuela Cercez

Ion și-a construit o vilă pe frumosul vârf al unui munte. Acum proiectează o scară specială, pe care va urca de la șosea până la vilă. Diferența de nivel dintre șosea și vilă este H (deci aceasta trebuie să fie înălțimea totală a scării).

Scara va avea N trepte, toate de aceeași lățime, dar de înălțimi distincte două câte două. *Ion* a sesizat că efortul pe care îl depune pentru a urca o treaptă este egal cu înălțimea treptei. Dar, dacă el urcă x trepte deodată, efortul depus este egal cu media aritmetică a înălțimilor acestor x trepte pe care le urcă deodată plus un efort de valoare constantă p (necesar pentru a-și lua avânt).

Fiind un tip athletic, *Ion* poate urca mai multe trepte deodată, dar suma înălțimilor treptelor urcate deo-

dată nu trebuie să depășească o valoare maximă M .

Scrieți un program care să determine efortul minim necesar pentru a urca pe o scară construită conform restricțiilor problemei, precum și o modalitate de a construi scara care va fi urcată cu efort minim.

Date de intrare

Fișierul de intrare **scara.in** conține pe prima linie 4 numere naturale separate prin câte un spațiu H, N, M și p (cu semnificația din enunț).

Date de ieșire

Fișierul de ieșire **scara.out** va conține pe prima linie efortul minim necesar (cu 2 zecimale exacte), iar pe cea de a doua linie N numere naturale nenule care reprezintă înălțimile celor N trepte ale scării (în ordinea de la șosea către vilă), separate prin câte un spațiu.

Restricții și precizări

Înălțimea scării (H) este cuprinsă între 1 și 75. Numărul de trepte (N) este cuprins între 1 și 8. Valoarea M este cuprinsă între 1 și 13. Valoarea p este cuprinsă între 1 și 13.

Dacă există mai multe soluții (modalități de a construi scara astfel încât să obținem efortul minim dorit), veți afișa prima soluție în ordine lexicografică. Spunem că vectorul $x = (x_1, x_2, \dots, x_k)$ precedă în ordine lexicografică vectorul $y = (y_1, y_2, \dots, y_k)$ dacă există $i \geq 1$ astfel încât $x_i = y_i$, pentru orice $j < i$ și $x_i < y_i$.

Dacă a doua zecimală a efortului minim este 0, sau chiar ambele zecimale sunt 0 nu este necesar să fie afișate. Așadar, pentru exemplul următor, s-ar fi putut scrie 9 sau 9.0.

Exemplu

scara.in

```
10 4 5 2
```

scara.out

```
9.00
1 4 2 3
```

Timp de execuție/test: 5 secunde

Clasele a XI-a și a XII-a

P050213: Lanț

Emanuela Cercez

Ion este un lingvist pasionat. Recent el a descoperit un text scris într-o limbă necunoscută.

Textul este scris pe mai multe linii și este format din cuvinte scrise cu litere mici din alfabetul latin, separate prin spații sau/și semne de punctuație (, : ; . ! ? -).

Ion a fost frapat că există multe similitudini între cuvintele din text. Fiind foarte riguros, *Ion* definește similitudinea a două cuvinte după cum urmează.

Fie c_1 și c_2 două cuvinte. Cuvântul c_1 poate fi obținut din cuvântul c_2 printr-o succesiune de operații elementare.

Operațiile elementare ce pot fi folosite sunt:

- **move** (c_1, c_2) - mută primul caracter din c_1 la sfârșitul cuvântului c_2 . Dacă $c_1 = \text{alba}$ și $c_2 = \text{neagra}$, după efectuarea operației **move** (c_1, c_2), c_1 va fi **lba**, iar c_2 va fi **neagraa**.
- **insert** (c_1, x) - Inserează caracterul x la începutul lui c_1 . Dacă $c_1 = \text{alba}$ și $x = b$, după efectuarea operației **insert** (c_1, x), c_1 va fi **balba**.
- **delete** (c_1) - Șterge primul caracter din c_1 . Dacă $c_1 = \text{alba}$, după efectuarea operației **delete** (c_1), c_1 va fi **lba**.

Definim similitudinea dintre c_1 și c_2 ca fiind numărul minim de operații **insert** și **delete** care trebuie să fie executate pentru a transforma cuvântul c_1 în cuvântul c_2 (operațiile **move** nu se numără).

Fie c_0 primul cuvânt din text. Începând cu c_0 putem construi lanțuri de k -similitudine.

Un lanț de k -similitudine este o succesiune de cuvinte distincte din text cu următoarele proprietăți:

- dacă cuvântul x apare în lanț înaintea cuvântului y , atunci prima apariție a lui x în text precedă prima apariție a lui y în text;
- dacă x și y sunt cuvinte consecutive în lanț (în ordinea $x y$), atunci similitudinea dintre x și y este k ;





- lanțul este maximal (adică nu putem adăuga încă un cuvânt la sfârșitul acestui lanț, astfel încât să fie respectate proprietățile precedente).

Scrieți un program care să determine numărul de lanțuri de k -similitudine care încep cu c_0 .

Date de intrare

Fișierul de intrare **lant.in** conține pe prima linie valoarea k . Pe următoarele linii se află textul dat.

Date de ieșire

Fișierul de ieșire **lant.out** va conține o singură linie pe care va fi scris numărul de lanțuri de k -similitudine care încep cu c_0 .

Restricții și precizări

Lungimea unei linii din text nu depășește 1000 de caractere. Lungimea unui cuvânt nu depășește 30 de caractere. Numărul total de cuvinte este mai mic sau egal cu 150.

Pentru datele de test, numărul de lanțuri de k -similitudine care încep cu c_0 va fi mai mic sau egal cu 2.000.000.000 (două miliarde).

Exemplu

lant.in

5

ana are mere, banane,
pere si castane.

lant.out

6

Explicație

Lanțurile de 5-similitudine care se pot forma sunt:

ana are mere pere
ana are pere
ana are banane castane
ana are si
ana banane castane
ana si

Timp de execuție/test: 1 secundă

P050214: Scara

Stelian Ciurea

Domnul G are de urcat o scară cu n trepte. În mod normal, la fiecare pas

pe care îl face, el urcă o treaptă. Pe k dintre aceste trepte se află câte o sticlă cu un număr oarecare de deciltri de apă, fie acesta x .

Dacă bea toată apa dintr-o astfel de sticlă, forța și mobilitatea lui G cresc, astfel încât, la următorul pas el poate urca până la x trepte, după care, dacă nu bea din nou ceva, revine la "normal". Sticlele cu apă nu costă nimic. Cantitatea de apă conținută de aceste sticle poate să difere de la o treaptă la alta. Pe j trepte se află câte o sticlă cu băutura energizantă. Și pentru aceste sticle, cantitatea de băutura energizantă poate să difere de la o treaptă la alta.

Să presupunem că într-una dintre aceste sticle avem y deciltri de băutura energizantă. Dacă bea q ($q \leq y$) deciltri dintr-o astfel de sticlă, la următorul pas G poate urca până la $2 \cdot q$ trepte, după care și în acest caz, dacă nu bea din nou ceva, el revine la "normal". Totuși băutura energizantă costă: pentru o cantitate de q deciltri consumați, G trebuie să plătească q lei.

Pot exista trepte pe care nu se află nici un pahar, dar și trepte pe care se află atât o sticlă cu apă cât și una cu băutura energizantă. În astfel de situații, nu are rost ca G să bea ambele băuturi deoarece efectul lor nu se cumulează; el poate alege să bea una dintre cele două băuturi sau poate să nu bea nimic.

Determinați p , numărul minim de pași pe care trebuie să îi facă G pentru a urca scara, precum și suma minimă pe care trebuie să o cheltuiască G pentru a urca scara în p pași.

Date de intrare

Fișierul text de intrare **scara.in** conține pe prima linie un număr natural n , reprezentând numărul total de trepte. Pe cea de a doua linie se află un număr natural k , reprezentând numărul de trepte pe care se află sticle cu apă. Pe fiecare dintre următoarele k linii se află câte două numere naturale separate printr-un spațiu, reprezentând numărul de ordine al treptei pe care se află o sticlă cu apă și respectiv cantitatea de apă din acea sticlă exprimată în deciltri.

Pe următoarea linie se află un număr natural j , reprezentând numărul de trepte pe care se află sticle cu băutura energizantă. Pe fiecare dintre următoarele j linii câte două numere naturale separate printr-un spațiu, reprezentând numărul de ordine al treptei pe care se află o sticlă cu băutura energizantă și respectiv cantitatea de băutura energizantă din acea sticlă exprimată în deciltri.

Date de ieșire

Fișierul text de ieșire **scara.out** va conține o singură linie pe care vor fi scrise două numere naturale p și c separate printr-un spațiu, care reprezintă numărul minim de pași, respectiv suma minimă.

Restricții și precizări

Numărul treptelor (n) este cuprins între 1 și 120. Numărul sticlelor cu apă este mai mic sau egal cu n . Numărul sticlelor cu băutura energizantă este mai mic sau egal cu n . Cantitatea de apă aflată în oricare sticlă este cuprinsă între 1 și 100. Cantitatea de băutura energizantă aflată în oricare sticlă este cuprinsă între 1 și 100.

Exemple

scara.in

6

1

1 2

2

4 1

1 2

scara.out

3 2

scara.in

6

1

1 2

2

4 1

1 1

scara.out

4 1

Timp de execuție/test: 1 secundă