



Olimpiada NAȚIONALĂ de Informatică 2004

Vă prezentăm în continuare enunțurile celor 18 de probleme propuse spre rezolvare la ediția din acest an a Olimpiadei Naționale de Informatică pentru elevii de gimnaziu și cele pentru lărgirea lotului național.

Clasa a V-a

P040501: Produs

Ionică, elev în clasa a V-a la o școală generală din Buzău este pasionat de matematică și de telefoane mobile. Într-un moment de plictiseală când se uită în agenda telefonică la un număr de telefon format din maxim 10 cifre (prima cifră fiind diferită de 0) se gândește la următoarea problemă: *Între care cifre ale numărului de telefon este bine să pună operatorul de înmulțire pentru a obține două numere cu produsul maxim?*

Se cere să se afișeze cele două numere și produsul lor.

Date de intrare

De la *intrarea standard* (tastatură) se va citi numărul de telefon.

Date de ieșire

La *ieșirea standard* (ecran) se vor furniza cele două numere și valoarea produsului unul sub altul. Cele două numere se vor afișa în ordinea cerută.

Restricții și precizări

- numărul de telefon este un număr natural cu cel puțin două cifre și mai mic sau egal cu 2100000000.

Exemplu

| <i>Intrare standard</i> | <i>Ieșirea standard</i> |
|-------------------------|-------------------------|
| 2301 | 2 301 602 |

Timp de execuție: 1 secundă/test

P040502: Doar o notă să-ți mai pun...

Astăzi, doamna profesoară va trece în catalog notele obținute la lucrarea de control de către elevii săi din clasa a V-a. Înainte de acest lucru dorește să afle care este nota (notele)

care apare de cele mai multe ori (are cea mai mare frecvență) și notele care nu s-au acordat la această lucrare.

Se cere să se determine și să se afișeze notele care apar de cele mai multe ori, precum și notele care nu s-au obținut la lucrare de către nici un elev.

Date de intrare

De la *intrarea standard* (tastatură) se va citi un număr natural n , care reprezintă numărul de elevi din clasă și cele n note obținute de elevi la lucrarea de control.

Datele vor fi introduse în ordinea specificată mai sus, iar după fiecare valoare numerică se va tasta <Enter>.

Date de ieșire

La *ieșirea standard* (ecran) se vor furniza pe o linie notele care apar de cele mai multe ori, separate între ele prin spațiu. Pe următoarea linie se vor furniza notele care nu s-au obținut la lucrare de către nici un elev, separate între ele prin spațiu.

Nu se vor afișa alte valori în afară de cele cerute.

Restricții și precizări

- $1 \leq n \leq 25$;
- toate notele sunt numere naturale cuprinse între 1 și 10.

Exemplu

| <i>Intrare standard</i> | <i>Ieșirea standard</i> |
|-------------------------|-------------------------|
| 4 | 5 |
| 4 | 1 2 3 6 8 9 10 |
| 5 | |
| 7 | |
| 5 | |

Timp de execuție: 1 secundă/test

P040503: Prințesa

În regatul numerelor naturale, prințesei *Prima* i-a venit sorocul să se mărite. Solii regelui au dat de veste că orice pre-



tendent la mâna prințesei trebuie să fie număr prim. Sfetnicii regelui au găsit însă H rude ale prințesei care nu pot să-i devină soț.

Cunoscându-se care sunt aceste rude, se cere să se afle cei dintâi K pretendenți posibili, în ordinea descrescătoare a lor.

Date de intrare

De la *intrarea standard* (tastatură) se vor citi numerele K și H și rudele prințesei. Datele se vor citi în ordinea precizată anterior, fiecare valoare numerică fiind urmată de <Enter>.

Date de ieșire

La *ieșirea standard* (ecran) se vor furniza cei K pretendenți, pe același rând, separați între ei printr-un spațiu.

Restricții

- $1 \leq H \leq 10$;
- $1 \leq K \leq 30$;
- rudele prințesei sunt numere naturale mai mici sau egale cu 100.

Exemplu

| <i>Intrare standard</i> | <i>Ieșirea standard</i> |
|-------------------------|-------------------------|
| K=4 | 13 11 5 2 |
| H=5 | |
| 4 | |
| 3 | |
| 12 | |
| 29 | |
| 7 | |

Timp de execuție: 1 secundă/test

Clasa a VI-a

P040504: Spider

Spider este un păianjen care trăiește în casa unui programator. De la acesta *Spider* a preluat pasiunea pentru numere și pentru programe. Astfel stând lucrurile, *Spider* a hotărât să nu-și mai țese pânza în mod tradițional, ci să folosească informațiile aflate de la programator, abordând și un stil de lucru metodic. Prin urmare, *Spider* procedează astfel:

- alege n puncte așezate în cerc și le numerează de la 1 la n (în sensul acelor de ceasornic);
- calculează distanțele dintre oricare două puncte obținând doar numere naturale distincte;
- alege un punct de plecare k ;
- stabilește următoarea regulă pe care o va respecta când va țese pânza: în fiecare zi va țese câte un fir: dacă numărul zilei este impar, atunci țese firul de la punctul în care se află la punctul următor (de asemenea în sensul acelor de ceasornic, iar după punctul numerotat cu n urmează punctul numerotat cu 1), iar dacă numărul zilei este par *Spider* țese un fir între punctul în care se află și punctul

în care ajunge sărind un punct;

- se oprește atunci când ar trebui să țese un fir între două puncte între care există deja un fir țesut.

Se cere să se determine numărul de zile necesar pentru a-și țese pânza și punctul în care s-a oprit și lungimile firelor țesute împreună cu capetele lor, în ordinea descrescătoare a lungimilor lor. Capetele firelor vor fi afișate în ordine crescătoare.

Date de intrare

Datele se citesc de la *intrarea standard* (tastatură).

Pe prima linie se află un număr natural n , reprezentând numărul de puncte alese.

Pe cea de-a doua linie se află un număr natural k , reprezentând punctul de plecare.

Pe fiecare dintre următoarele n linii se află n numere naturale separate între ele prin spații, astfel că, cel de-al j -lea element de pe cea de-a i -a linie (d_{ij}) reprezintă distanța dintre punctele numerotate cu i , respectiv j găsită de *Spider*.

Date de ieșire

La *ieșirea standard* (ecran) se vor scrie pe prima linie două numere x și p , separate printr-un spațiu, care reprezintă numărul de zile, respectiv punctul în care s-a oprit *Spider*.

Pe următoarele p linii se vor scrie trei numere l , $c1$, $c2$ separate între ele prin spațiu, care reprezintă lungimea unui fir și capetele lui. Capetele firelor se vor scrie în ordine crescătoare, iar datele pe aceste p linii se vor scrie în ordine descrescătoare a lungimilor.

Restricții și precizări

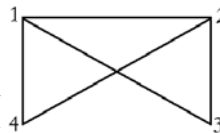
- $1 \leq n \leq 100$;
- $1 \leq k \leq n$;
- $0 \leq d_{ij} \leq 50.000, \forall 1 \leq i, j \leq n$.

Exemplu

| <i>Intrare standard</i> | <i>Ieșirea standard</i> |
|-------------------------|-------------------------|
| 4 | 5 1 |
| 2 | 10 2 4 |
| 0 5 8 7 | 8 1 3 |
| 5 0 3 10 | 7 1 4 |
| 8 3 0 4 | 5 1 2 |
| 7 10 4 0 | 3 2 3 |

Explicație

În prima zi *Spider* pleacă din punctul 2 și țese un fir până la punctul 3. În ziua a doua, din punctul 3 țese un fir până la punctul 1 (sare punctul 4). În ziua a treia din punctul 1 țese un fir până la punctul 2. În ziua a patra din punctul 2 țese un fir până la punctul 4 (sare punctul 3). În ultima zi din punctul 4 țese un fir până la punctul 1 și se oprește.



Timp de execuție: 1 secundă/test



P040505: Numere prime "apropiate"

Pentru temă, *Alina* trebuia să-și verifice cunoștințele legate de numere prime.

Alina va citi (chiar în această ordine) o valoare N care semnifică numărul numerelor naturale care vor fi prelucrate și o valoare C care poate fi 1 sau 2. În continuare va citi cele N numere naturale. Fiecare număr natural A citit, va fi prelucrat astfel:

- dacă A este număr prim, el va fi afișat nemodificat;
- dacă A nu este prim, se va determina cel mai apropiat număr prim care va fi afișat. Considerăm cel mai apropiat număr de valoarea A , cel pentru care modulul diferenței dintre număr și valoarea A este cel mai mic. Fie X cel mai mare număr prim mai mic decât A , și Y cel mai mic număr prim mai mare decât A . Dacă cele două numere sunt la fel de "apropiate" de A atunci, dacă $C = 1$ se va afișa X și dacă $C = 2$ se va afișa Y .

Date de intrare

Datele se citesc de la *intrarea standard* (tastatură). Pe prima linie se află două numere N și C , separate între ele printr-un singur spațiu, reprezentând numărul de numere care trebuie prelucrate, respectiv valoarea C care poate fi 1 sau 2, valoare care va decide dacă se afișează valoarea primă cea mai apropiată mai mică, respectiv mai mare decât valoarea corespunzătoare.

Pe cea de-a doua linie se află cele N numere (a_1, a_2, \dots, a_n) care trebuie prelucrate, separate între ele prin spațiu.

Date de ieșire

La *ieșirea standard* (ecran) se vor afișa N numere naturale, separate printr-un spațiu, reprezentând numerele prime "apropiate" de cele inițiale.

Restricții și precizări

- $1 \leq N \leq 150$;
- $3 \leq a_i \leq 32.700$;
- nu se vor afișa pe ecran alte valori, în afara celor cerute ca rezultate ale problemei.

Exemple

Intrare standard

15 1
3 6 8 2 3 5 7 9 2 5 3 7 11 22 21

Ieșirea standard

3 5 7 2 3 5 7 7 2 5 3 7 11 23 19

Intrare standard

13 2
3 6 7 11 21 9 5 7 14 29 24 25 16

Ieșirea standard

3 7 7 11 23 11 5 7 13 29 23 23 17

Timp de execuție: 1 secundă/test

P040506: Palindrom

Ionel are de rezolvat la matematică o problemă care presupune calcularea unei expresii care conține paranteze, operații de adunare, scădere, înmulțire, împărțire cu numere naturale.

Problema este că *Ionel* nu are chef să calculeze, astfel încât a început să se joace cu toate numerele din expresie: le-a despărțit în cifrele componente și, fiindcă a auzit de numere palindroame de la un coleg care știa ceva informatică, a încercat să scrie un număr cât mai mare cu cifrele pe care le are la dispoziție și care să fie palindrom.

Deoarece jocul i s-a părut prea simplu a decis ca în cazul în care va folosi o cifră el va trebui să o utilizeze de câte ori apare în expresia aritmetică.

Cunoscând cifrele din expresie, să se determine cel mai mare număr palindrom ce se poate forma respectând regula de mai sus.

Un număr este palindrom dacă citit de la stânga la dreapta are aceeași valoare ca și la citirea de la dreapta la stânga (exemplu: 17271).

Date de intrare

Datele se citesc de la *intrarea standard* (tastatură). Pe prima linie se află un număr natural n care reprezintă numărul de cifre. Pe cea de-a doua linie se află cele n cifre separate între ele printr-un spațiu.

Date de ieșire

La *ieșirea standard* (ecran) se va afișa, pe o singură linie, numărul cerut.

Restricții și precizări

- $1 \leq n \leq 10.000$;
- nu se vor afișa alte valori numerice pe ecran, în afara numărului cerut și nu se vor afișa spații între cifrele numărului.

Exemple

| | |
|-------------------------|-------------------------|
| <i>Intrare standard</i> | <i>Ieșirea standard</i> |
| 4 | 252 |
| 2 4 5 2 | |

| | |
|-------------------------|-------------------------|
| <i>Intrare standard</i> | <i>Ieșirea standard</i> |
| 7 | 333 |
| 2 2 3 5 3 3 2 | |

Timp de execuție: 1 secundă/test

Clasa a VII-a

P040507: V-uri

Se consideră un tablou bidimensional care are m linii și n coloane. Se numește traseu în V o parcurgere prin elementele tabloului astfel:

- se pleacă întotdeauna de la un element de pe linia 1 a tabloului, se ajunge în final la un alt element de pe linia 1 a



tabloului, trecând prin cel puțin trei elemente, fără a trece printr-un element de mai multe ori;

- parcurgerea elementelor tabloului se face în forma unei singure litere *V* ca în figura din exemplu, de la element putându-se trece doar la un alt element imediat vecin pe diagonală.

Fiecare element al tabloului conține valori întregi. La parcurgerea traseului se calculează suma elementelor de pe traseu.

Determinați traseul în *V* care conține cea mai mare sumă.

În cazul în care există mai multe trasee cu aceeași sumă, se va alege traseul care parcurge cele mai puține celule. Dacă și în acest caz există mai multe soluții, se alege traseul cel mai din stânga (cel cu indicele coloanei de pornire cel mai mic).

Date de intrare

Fișierul de intrare **v.in** conține pe prima linie două numere naturale *m* și *n*, separate între ele printr-un singur spațiu, reprezentând numărul de linii și numărul de coloane ale tabloului.

Pe următoarele *m* linii se află câte *n* numere, separate între ele prin spațiu, care reprezintă valorile elementelor unei linii din tablou.

Date de ieșire

Fișierul de ieșire **v.out** va conține o singură linie pe care se vor afla trei numere naturale, separate între ele prin spațiu, reprezentând suma valorilor elementelor din tablou pentru traseul ales, coloana de pornire și linia pe care se află vârful *V*-ului.

Datele din fișierul de ieșire trebuie să fie în ordinea specificată, adică sumă, coloană, linie.

Restricții și precizări

- $1 \leq m, n \leq 100$;
- valorile elementelor tabloului sunt numere întregi cuprinse între -60.000 și 60.000;
- conform datelor de intrare, suma valorilor elementelor din tablou pentru orice traseu ales nu depășește 1.000.000.000.

Exemplu

v.in

```
5 9
3 4 12 4 6 7 9 5 12
0 4 5 7 9 -5 1 1 5
0 98 34 0 1 7 7 1 1
6 7 8 -9 0 2 3 5 22
47 62 31 55 0 83 23 77 10
```

v.out

```
54 1 3
```

Conform datelor de mai sus se pot crea și alte trasee, dar care au suma mai mică decât cea din rezultat:

| | | | | | | | | |
|----|----|----|----|---|----|----|----|----|
| 3 | 4 | 12 | 4 | 6 | 7 | 9 | 5 | 12 |
| 0 | 4 | 5 | 7 | 9 | -5 | 1 | 1 | 5 |
| 0 | 98 | 34 | 0 | 1 | 7 | 7 | 1 | 1 |
| 6 | 7 | 8 | -9 | 0 | 2 | 3 | 5 | 22 |
| 47 | 62 | 31 | 55 | 0 | 83 | 23 | 77 | 10 |

Timpi de execuție: 1 secundă/test

P040508: Jetoane

Ionel este elev în clasa a IV-a și părinții s-au gândit la o metodă eficientă prin care el să învețe numerele mari și modul lor de formare. În acest scop i-au cumpărat un joc cu jetoane. Pentru fiecare cifră există câte 10 jetoane inscripționate pe una din fețe cu cifra respectivă. *Ionel* poate forma numere prin așezarea jetoanelor unul lângă altul. Părinții îi cer lui *Ionel* să formeze numere, pe rând, care să aibă suma cifrelor *S* și să fie mai mici decât 10^a . Deoarece este dificil să-l urmărească în formarea numerelor, părinții doresc să știe câte astfel de numere distincte știe *Ionel* să formeze.

Câte numere distincte a format *Ionel*?

Date de intrare

Fișierul de intrare **jetoane.in**, conține pe prima linie numărul *S*, iar pe cea de-a doua linie numărul *a*.

Date de ieșire

În fișierul **jetoane.out** se va scrie pe un singur rând numărul cerut.

Restricții și precizări

- $1 \leq S < 10$;
- $1 \leq a < 10$;
- *S* și *a* sunt numere naturale.

Exemple

| jetoane.in | jetoane.out |
|------------|-------------|
| 2 | 6 |
| 3 | |

| jetoane.in | jetoane.out |
|------------|-------------|
| 5 | 126 |
| 5 | |

Explicație

În cazul primului exemplu (*S* = 2 și *a* = 3) *Ionel* formează numerele: 2, 11, 20, 101, 110, 200.

Timpi de execuție: 1 secundă/test

P040509: Neo

Neo, eroul din filmul *Matrix*, pentru a-și învinge adversarii are nevoie de energie. El a descoperit în matrice o zonă care

cuprinde locații consecutive numerotate de la 1 la N , fiecare dintre ele generând o anumită cantitate de energie, pozitivă sau negativă, cunoscută.

Neo are posibilitatea să pătrundă în această zonă printr-o singură locație și poate părăsi zona o singură dată. El, odată intrat în zona energetică, trebuie să străbată locațiile în ordinea crescătoare a numărului lor, iar energia acumulată într-o locație multiplică de atâtea ori energia deja acumulată de *Neo*.

De asemenea, dacă într-o locație se află energie negativă, iar *Neo* a acumulat până în acea locație energie pozitivă, toată energia lui se transformă în energie negativă, dar dacă în drumul său *Neo* întâlnește o altă locație cu energie negativă, iar energia deținută de el este tot negativă, toată energia lui devine pozitivă.

Energia inițială a lui *Neo* are valoarea 1.

Ajutați-l pe *Neo* să descopere locația în care trebuie să intre și locația în care trebuie să părăsească zona energetică pentru a acumula cea mai mare cantitate de energie pozitivă posibilă.

Date de intrare

Fișierul de intrare **NEO.IN** conține pe prima linie numărul locațiilor energetice, iar pe cea de-a doua linie conține un șir de numere întregi, separate între ele printr-un spațiu, reprezentând cantitățile de energie ale locațiilor.

Date de ieșire

Pe prima linie a fișierului de ieșire **NEO.OUT** se va scrie cantitatea de energie acumulată de *Neo*, iar pe cea de-a doua linie se vor scrie două numere separate între ele printr-un singur spațiu, care reprezintă numărul de ordine al locației în care *Neo* pătrunde în zona energetică și numărul de ordine al locației în care *Neo* părăsește zona energetică. Dacă nu se poate obține energie pozitivă, cantitatea de energie acumulată și numărul de ordine al locațiilor vor fi 0.

Restricții și precizări

- $0 \leq N \leq 100$;
- valoarea energiei dintr-o locație este un număr întreg cuprins între -3 și 3;
- produsul tuturor energiilor este un număr întreg cuprins între -2.000.000.000 și 2.000.000.000;
- dacă există mai multe soluții pentru datele de intrare, se va afișa una singură.

Exemple

| NEO.IN | NEO.OUT |
|------------|---------|
| 4 | 12 |
| -3 -2 2 -3 | 2 4 |

| NEO.IN | NEO.OUT |
|--------|---------|
| 1 | 0 |
| -2 | 0 0 |

Timpi de execuție: 1 secundă/test

Clasa a VIII-a

P040510: Zar

Pe tabla de șah de dimensiune $N \times N$, se găsește un zar (care are latura egală cu latura pătratului care definește un pătrat al tablei de șah). Acest zar se poate rostogoli în conformitate cu secvența de comenzi care se citește dintr-un fișier, fiecare caracter citit având semnificația:

- u – rostogolire în pătratul de deasupra (*up*);
- d – rostogolire în pătratul de dedesubt (*down*);
- l – rostogolire în pătratul din stânga (*left*);
- r – rostogolire în pătratul din dreapta (*right*).

Se cunosc:

- poziția inițială a zarului dată prin punctul de coordonate (linie, coloană);
- valorile aflate pe fețele zarului date printr-un șir: *sus, jos, stânga, dreapta, față, spate*, în această ordine;
- secvența de mișcări date este formată din caracterele 'u', 'd', 'l' și 'r'.

Să se determine sumele de valori pe care le vor avea, în urma secvenței de mișcări, fețele: *sus, jos, stânga, dreapta, față, spate* (în această ordine), precum și mulțimea valorilor care sunt pe fața de sus a zarului la fiecare mutare a acestuia.

Prima poziție se consideră cea inițială.

Secvența de mișcări a zarului se realizează cât timp zarul printr-o mutare rămâne pe tabla de șah. Dacă în urma unei mutări zarul ar ieși în afara tablei, se vor afișa rezultatele obținute până în acel moment.

Date de intrare

Fișierul de intrare **ZAR.IN** conține pe prima linie un număr natural N care reprezintă dimensiunea tablei.

Cea de-a doua linie conține două numere lp și cp , separate între ele printr-un singur spațiu, care reprezintă linia inițială, respectiv coloana inițială pentru poziția zarului.

Cea de-a treia linie conține șase numere v_1, v_2, v_3, v_4, v_5 și v_6 , separate între ele printr-un spațiu, care reprezintă valorile fețelor zarului și sunt cuprinse între 1 și 6.

Cea de-a patra linie conține un număr natural b care reprezintă numărul de caractere ale secvenței de mutări, iar ultima linie conține caracterele secvenței de mișcări ne-separate prin spații.

Date de ieșire

Fișierul de ieșire cu numele **ZAR.OUT** va trebui să conțină pe prima linie șase numere *suma_sus, suma_jos, suma_stanga, suma_dreapta, suma_fata, suma_spate*, separate între ele prin spațiu, cu semnificația suma punctelor de pe fețele specificate, în această ordine.

Pe cea de-a doua linie a fișierului de ieșire se vor afla $b + 1$ valori care reprezintă valorile de pe fața de sus a zarului pentru secvența de mișcări propusă. Pe prima poziție se va afla valoarea inițială.





Restricții și precizări

- $4 \leq N \leq 100$;
- secvența de mișcări poate avea până la 1.000 de caractere.

Exemple

ZAR.IN

```
8
7 3
1 6 5 2 4 3
19
uuuuuurrrrrdddlldduu
```

ZAR.OUT

```
72 68 77 63 67 73
1 4 6 3 1 4 6 5 1 2 6 4 1 3 2 4 6 3 6 4
```

ZAR.IN

```
8
7 3
1 6 5 2 4 3
9
uuuuuuuuu
```

ZAR.OUT

```
25 24 35 14 27 22
1 4 6 3 1 4 6
```

Timpi de execuție: 1 secundă/test

P040511: Sume

Fie n un număr natural nenul. Scrieți un program care să determine n submulțimi disjuncte două câte două de câte n elemente distincte din mulțimea $\{1, 2, \dots, n^2\}$, submulțimi pentru care suma elementelor este aceeași.

Date de intrare

Fișierul de intrare **sume.in** conține pe prima linie numărul natural nenul n .

Date de ieșire

Fișierul de ieșire **sume.out** va conține n linii, câte una pentru fiecare submulțime determinată. Pe linia i se află cele n elemente ale submulțimii i , separate prin câte un spațiu.

Restricții și precizări

- $1 \leq n \leq 100$;
- două submulțimi sunt disjuncte dacă nu au elemente comune;
- soluția nu este unică, puteți afișa orice soluție care respectă condițiile din enunțul problemei;
- ordinea submulțimilor sau a elementelor submulțimii nu contează.

Exemplu

```
sume.in      sume.out
4            11 6 1 16
```

```
15 10 5 4
3 8 9 14
13 2 7 12
```

Timpi de execuție: 1 secundă/test

P040512: Plăci meteoritice

Pe planeta *UZABU* va cădea o ploaie de meteoriți. Savanții știu că fiecare meteorit este de formă dreptunghiulară, având laturile paralele cu axele de coordonate (și pe planeta *UZABU* axele de coordonate au aceeași semnificație ca și pe Pământ).

Solul planetei este reprezentat prin axa *OX*. Meteoriții căzuți pe planetă sunt folositori pentru agricultură. Dacă un meteorit în cădere atinge un alt meteorit, atunci amândoi se vor distruge, iar craterul format distruge solul planetei.

Cunoscând care sunt coordonatele plăcilor date prin patru numere (x_1, y_1, x_2, y_2) , cu semnificația: (*stânga_sus_x*, *stânga_sus_y*, *dreapta_jos_x*, *dreapta_jos_y*), savanții trebuie să distrugă o parte dintre meteoriți astfel încât ei să nu se suprapună în momentul atingerii solului.

Date de intrare

Fișierul de intrare **METEOR.IN** conține pe prima linie un număr natural N reprezentând numărul de meteoriți.

Pe fiecare dintre următoarele N linii se află câte patru numere separate între ele printr-un spațiu, reprezentând coordonatele *stânga sus* și *dreapta jos* ale unei plăci de formă dreptunghiulară.

Date de ieșire

Fișierul de ieșire **METEOR.OUT** va conține o singură valoare k , reprezentând numărul maxim al plăcilor rămase.

Restricții și precizări

- $0 \leq N \leq 500$;
- coordonatele fiecărui meteorit sunt numere întregi cuprinse între 0 și 32.000;
- două plăci care prin cădere se lipesc nu se distrug.

Exemplu

```
METEOR.IN      METEOR.OUT
10              6
5 10 18 2
15 22 27 12
35 30 40 25
43 30 45 25
32 20 45 12
50 20 60 15
20 30 30 25
65 20 82 15
48 13 75 2
78 14 100 3
```

Timpi de execuție: 1 secundă/test

P040513: Invsort

Se consideră un șir de N numere naturale, care trebuie ordonat crescător. Singura operație permisă este să considerăm elementele de pe pozițiile $i, i+1, \dots, j$ (pentru i și j arbitrare, $i < j$), și să inversăm ordinea acestor elemente (adică elementul de pe poziția i ajunge pe poziția j , $i+1$ ajunge pe poziția $j-1$, ..., j ajunge pe poziția i). Costul unei astfel de operații este numărul de elemente din subșirul inversat, și anume $j-i+1$.

Scrieți un program care să determine o secvență de operații care ordonează crescător șirul dat și are un cost total cât mai mic (dar nu obligatoriu minim).

Date de intrare

Fișierul de intrare **invsort.in** conține pe prima linie numărul N și apoi N linii cu elementele șirului dat (nu neapărat distincte).

Date de ieșire

Fișierul de ieșire **invsort.out** va conține pe fiecare linie descrierea unei operații. O operație este descrisă prin două numere i și j , separate printr-un spațiu. Aceste numere satisfac $1 \leq i < j \leq N$.

Restricții și precizări

- $2 \leq N \leq 32.000$;
- valorile șirului care trebuie ordonat sunt între 0 și 31.999.

Exemplu

| invsort.in | invsort.out |
|-------------------|--------------------|
| 5 | 3 5 |
| 1 | 1 3 |
| 0 | |
| 1 | |
| 1 | |
| 0 | |

Explicație

- prima operație are efectul: $1\ 0\ [1\ 1\ 0] \rightarrow 1\ 0\ 0\ 1\ 1$
- a doua operație are efectul: $[1\ 0\ 0]\ 1\ 1 \rightarrow 0\ 0\ 1\ 1\ 1$
- costul total este $3 + 3 = 6$

Timpi de execuție:

- *Microsoft Windows*: 2 secunde/test
- *Linux*: 0,5 secunde/test

P040514: Peri

Se consideră o matrice dreptunghiulară A cu m linii și n coloane cu valori 0 sau 1, liniile și coloanele fiind numerotate de la 1 la m , respectiv de la 1 la n . Numim dreptunghi de colțuri $(x_1, y_1) (x_2, y_2)$ cu $x_1 < x_2$ și $y_1 < y_2$ mulțimea elementelor A_{ij} cu $x_1 \leq i \leq x_2$ și $y_1 \leq j \leq y_2$. Numim perimetru al dreptunghiului de colțuri $(x_1, y_1) (x_2, y_2)$ mulțimea elementelor A_{ij} pentru care $(i = x_1 \text{ și } y_1 \leq j \leq y_2)$ sau $(i = x_2 \text{ și } y_1 \leq j \leq y_2)$ sau $(i = x_1 \text{ și } y_1 \leq j \leq y_2)$ sau $(i = x_2 \text{ și } y_1 \leq j \leq y_2)$.

$y_1 \leq j \leq y_2$) sau $(j = y_1 \text{ și } x_1 \leq i \leq x_2)$ sau $(j = y_2 \text{ și } x_1 \leq i \leq x_2)$.

Determinați diferența maximă dintre numărul de elemente egale cu 1 și numărul de elemente egale cu 0 aflate pe perimetrul aceluiași dreptunghi, precum și numărul de dreptunghiuri pentru care se obține această diferență.

Date de intrare

Pe prima linie a fișierului de intrare **peri.in** sunt scrise numerele m și n , separate printr-un singur spațiu. Pe următoarele m linii este dată matricea A , numerele de pe aceeași linie fiind separate de câte un spațiu.

Date de ieșire

Fișierul de ieșire **peri.out** va conține o singură linie pe care se află două numere întregi separate printr-un spațiu. Primul număr este diferența maximă dintre numărul de elemente 1 și numărul de elemente 0 de pe perimetrul unui dreptunghi. Al doilea întreg este numărul de dreptunghiuri pentru care diferența dintre numărul de elemente 1 și numărul de elemente 0 de pe perimetru este maximă.

Restricții și precizări

- $1 < m, n \leq 250$;
- prin diferență nu se înțelege diferență în valoare absolută.

Exemplu

| peri.in | peri.out |
|----------------|-----------------|
| 4 5 | 4 2 |
| 1 0 0 1 0 | |
| 0 1 1 0 0 | |
| 0 1 0 1 0 | |
| 1 1 1 0 1 | |

Timpi de execuție:

- *Microsoft Windows*: 1 secundă/test
- *Linux*: 0,2 secunde/test

P040515: Trans

În depozitul unei companii de construcții se află N blocuri de piatră, de culoare albă sau neagră. Ele sunt așezate în ordinea 1, 2, ..., N , de la intrarea în depozit către interior.

Blocurile de piatră trebuie să fie transportate pe un șantier de construcții, în ordinea în care ele sunt depozitate, iar pentru aceasta va trebui închiriat un camion de la o companie de transport. Aceasta deține Q tipuri de camioane. Camionul de tipul i ($1 \leq i \leq Q$) poate transporta maxim K_i blocuri de piatră la un moment dat și pentru un transport se percepe taxa T_i .

Compania de transport este de părere că, pentru a-și păstra clientela, trebuie să impună anumite standarde, indiferent de cât de absurde ar fi, deci impune condiția ca toate blocurile de piatră plasate în camion la un transport să aibă aceeași culoare. Așadar, pentru a fi transportate toate blocurile pe șantier, compania de construcții va alege un camion de un anumit tip, iar camionul va efectua unul sau mai multe transporturi.





Pentru a micșora suma totală plătită, compania de construcții are posibilitatea de a schimba culoarea oricărui bloc de piatră (din alb în negru sau din negru în alb); pentru fiecare bloc i ($1 \leq i \leq N$) se cunoaște suma S_i care trebuie plătită pentru a-i schimba culoarea.

Pentru fiecare dintre cele Q tipuri de camioane deținute de compania de transport, determinați suma minimă pe care o va plăti compania de construcții pentru a transporta toate cele N blocuri pe șantier.

Date de intrare

Fișierul de intrare **trans.in** conține pe prima linie numărul întreg N , reprezentând numărul de blocuri de piatră din depozit.

Pe fiecare dintre următoarele N linii se află informații referitoare la câte un bloc de piatră. Pe a i -a dintre aceste N linii se găsesc două numere întregi separate printr-un spațiu: C_i și S_i , reprezentând culoarea celui de-al i -lea bloc (C_i este 0 pentru alb și 1 pentru negru) și respectiv suma ce trebuie plătită pentru a-i schimba culoarea (dacă este necesar).

Pe următoarea linie se află numărul natural Q , reprezentând numărul de tipuri de camioane deținute de compania de transport. Pe fiecare dintre următoarele Q linii se află informații referitoare la câte un camion. Pe cea de a i -a dintre aceste Q linii sunt scrise două numere naturale, separate printr-un spațiu K_i T_i , reprezentând numărul maxim de blocuri care pot fi transportate simultan de către un camion de tipul i și respectiv taxa care trebuie plătită pentru fiecare transport efectuat.

Date de ieșire

Fișierul de ieșire **trans.out** va conține Q linii. Pe cea de a i -a dintre aceste linii va fi afișată suma totală minimă plătită de compania de construcții pentru a transporta toate cele N blocuri de piatră, în cazul în care ar închiria un camion de tipul i .

Restricții și precizări

- $1 \leq N \leq 16.000$;
- $1 \leq S_i \leq 10.000$;
- $1 \leq Q \leq 100$;
- $1 \leq K_i \leq N$;
- $1 \leq T_i \leq 100.000$.

Exemplu

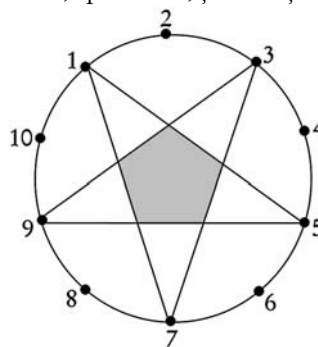
| trans.in | trans.out |
|-----------------|------------------|
| 4 | 1005 |
| 0 2 | 4 |
| 1 3 | 14 |
| 0 10 | |
| 1 2 | |
| 3 | |
| 4 1000 | |
| 4 1 | |
| 2 5 | |

Timpi de execuție:

- *Microsoft Windows*: 1 secundă/test
- *Linux*: 0,3 secunde/test

P050326: Poligon

Geo a învățat o metodă de a fixa n puncte pe un cerc de rază r , astfel încât să împartă cercul în n coarde egale ca lungime. Apoi și-a ales un număr k și a început să unească punctele succesiv, din k în k , păstrând același sens, până când a ajuns în punctul din care a pornit. Astfel, dacă a fixat $n = 10$ puncte pe cerc pe care le-a numerotat 1, 2, ..., 10 și și-a ales $k = 6$, atunci el unește punctul 1 cu 7, apoi pe 7 cu 3, apoi 3 cu 9, apoi 9 cu 5, și în sfârșit 5 cu 1.



Apoi a colorat poligonul format în interior, pornind din centrul cercului și fără a depăși vreuna dintre liniile desenate. El se întreabă în final câte laturi are poligonul colorat și care este aria acestuia.

Pentru numerele naturale date n , k și r , se cere numărul de laturi L ale poligonului colorat și aria S a acestuia (cu 2 zecimale exacte).

Date de intrare

Din fișierul **poligon.in** se citesc trei numere naturale n , k și r , despărțite prin câte un spațiu.

Date de ieșire

În fișierul **poligon.out** se vor scrie, pe prima linie numărul L de laturi ale poligonului colorat, iar pe linia a doua numărul real reprezentând aria acestuia.

Restricții și precizări

- $3 < n < 10.001$;
- $0 < k < n$;
- pentru n par, $2 \cdot k \neq n$;
- $10 < r < 501$;
- pentru fiecare test, dacă numărul de laturi determinat este corect, se acordă 20% din punctajul maxim de pe testul respectiv, iar dacă și aria determinată este corectă atunci se acordă punctajul maxim.

Exemple

| poligon.in | poligon.out |
|-------------------|--------------------|
| 10 6 100 | 5 |
| | 3468.92 |

Timpi de execuție:

- *Microsoft Windows*: 0,1 secunde/test
- *Linux*: 0,1 secunde/test

P040517: Poliție

Gigel și *Costel* sunt doi polițiști cu experiență. Ei lucrează împreună de mulți ani și de multe ori au fost nominalizați pentru premiul "Polițiștii anului". Anul acesta sunt hotărâți să-l câștige și pentru aceasta trebuie să încaseze cât mai mulți bani din amenzi.

În fiecare zi, *Gigel* și *Costel* pot aplica trei tipuri de amenzi:

- amendă de tipul 1, care se acordă pietonilor care traversează neregulamentar și pentru care se încasează suma S_1 și are durata de aplicare T_1 ;
- amendă de tipul 2, care se acordă șoferilor de autovehicule care încalcă regulile de circulație și pentru care se încasează suma S_2 și are durata de aplicare T_2 ;
- amendă de tipul 3, care se acordă șoferilor de mașini grele, cu transport ilegal de marfă și pentru care se încasează suma S_3 și are durata de aplicare T_3 .

Amenzile de tipul 1 și 2 pot fi aplicate de un singur polițist (*Gigel* sau *Costel*). Pentru o amendă de tipul 3, *Gigel* și *Costel* trebuie să lucreze împreună (unul verifică actele de transport, iar celălalt verifică marfa).

Durata de aplicare a unei amenzi reprezintă timpul necesar polițiștilor pentru a verifica acte, a scrie proces verbal etc. Dacă un polițist aplică o amendă la momentul x , iar durata aplicării amenzii este y , polițistul care aplică amenda va deveni disponibil abia la momentul $x + y$. Polițiștii nu fac minute suplimentare, ei sunt în activitate din minutul 1 până în minutul T exclusiv, deci trebuie să fie liberi să plece acasă în minutul T . Pentru că în noul cod rutier nu le mai este permis polițiștilor să tragă șoferii pe dreapta și să-i lase să aștepte, pentru a aplica o amendă de tipul 1 sau 2 trebuie să fie liber măcar un polițist, iar pentru a aplica o amendă de tipul 3 ambii polițiști trebuie să fie liberi la momentul în care survine evenimentul.

Cei doi polițiști stau la pândă și observă evenimentele din trafic. Dacă nu pot aplica amenzi pentru toate evenimentele care intervin în trafic, ei sunt nevoiți să le aleagă pe acelea care, în total, le aduc mai mulți bani.

Scrieți un program care să determine suma maximă pe care o pot încasa din amenzi *Gigel* și *Costel* într-o tură.

Date de intrare

Fișierul de intrare **politie.in** conține:

- pe prima linie numărul natural T , reprezentând minutul la care cei doi polițiști sunt liberi să plece acasă
- pe linia a doua, două numere naturale S_1 T_1 (suma încasată și durată aplicării unei amenzi de tipul 1);

- pe linia a treia, două numere naturale S_2 T_2 (suma încasată și durată aplicării unei amenzi de tipul 2);
- pe linia a patra, două numere naturale S_3 T_3 (suma încasată și durată aplicării unei amenzi de tipul 3);
- pe linia a cincea, un număr natural N (numărul de evenimente survenite în trafic);
- pe fiecare dintre următoarele N linii se află două numere naturale *tip timp* (*tip* poate fi 1, 2 sau 3 și reprezintă tipul amenzii care poate fi aplicată; *timp* reprezintă timpul la care a survenit evenimentul, exprimat în număr de minute față de începutul turei).

Numerele scrise pe aceeași linie sunt separate între ele prin câte un spațiu. Evenimentele sunt date în ordine cronologică.

Date de ieșire

Fișierul de ieșire **politie.out** conține o singură linie pe care este scrisă suma maximă care poate fi încasată.

Restricții și precizări

- $0 < T, T_1, T_2, T_3 < 201$;
- $0 < S_1, S_2, S_3 < 51$;
- $0 < N < 501$;
- există evenimente care intervin simultan în trafic.

Exemplu

| politie.in | politie.out |
|-------------------|--------------------|
| 300 | 140 |
| 10 20 | |
| 30 30 | |
| 50 25 | |
| 8 | |
| 1 5 | |
| 3 10 | |
| 3 20 | |
| 1 130 | |
| 2 142 | |
| 2 160 | |
| 3 180 | |
| 2 280 | |

Explicație

Aplică amândoi amenda de tipul 3 din minutul 10, apoi *Costel* aplică singur amenda de tipul 1 din minutul 130, apoi *Gigel* aplică singur amenda de tipul 2 din minutul 142, apoi amândoi aplică amenda de tipul 3 din minutul 180.

Timpi de execuție:

- *Microsoft Windows*: 1,2 secunde/test
- *Linux*: 0,2 secunde/test

P040518: Sea

Pe mare se află N vapoare. Malul este în mod curios perfect drept și este reprezentat prin axa Ox a sistemului de coor-





donate. Cele N vapoare sunt reprezentate prin perechi de coordonate (Vx_i, Vy_i) , unde Vy_i este strict pozitiv (marea este deasupra axei Ox). Pe mal se află M faruri, date prin coordonatele lor Fx_i (fiind exact la limita dintre mare și uscat, coordonata y a lor este întotdeauna 0). Cele M faruri sunt ciudate pentru că ele nu pot lumina decât în stânga. Astfel aria luminată de fiecare far i este delimitată de un sfert de cerc cu o rază Fri . Mai exact, un vapor este luminat de un anumit far dacă se află în stânga farului (are coordonata x mai mică) și distanța de la far la vapor este mai mică sau egală cu valoarea Fri asociată farului respectiv.

Pentru fiecare far se mai dă și un număr natural strict pozitiv Fn_i . Din motive greu de înțeles, șeful portului dorește ca fiecare far i să lumineze cel puțin Fn_i vapoare (un vapor poate fi luminat de mai multe faruri). El dorește consum minim de energie și vrea să afle pentru fiecare far raza minimă necesară pentru a lumina numărul cerut de vapoare.

Determinați pentru fiecare far valoarea Fri care reprezintă raza minimă necesară pentru ca farul să lumineze cel puțin Fn_i vapoare.

Date de intrare

Prima linie a fișierului **sea.in** conține două numere întregi N și M , separate printr-un spațiu, reprezentând numărul de vapoare, respectiv numărul de faruri.

Fiecare dintre următoarele N linii conține câte o pereche de numere reale separate printr-un spațiu Vx_i și Vy_i (coordoanatele vapoarelor).

Fiecare dintre următoarele M linii conține câte o pereche de numere separate printr-un spațiu, unul real Fx_i și unul întreg Fn_i (coordoanatele orizontale și numerele asociate farurilor).

Date de ieșire

Fișierul **sea.out** va conține M linii, fiecare linie conținând un număr real, dat cu patru zecimale: pe linia i se află raza minimă necesară pentru ca farul i să lumineze Fn_i vapoare.

Restricții

- $1 \leq N \leq 400$, $1 \leq M \leq 100.000$;
- $0 < y, r < 100.000$;
- $-100.000 < x < 100.000$;
- $1 \leq Fn_i \leq N$;
- în fișierul de intrare farurile sunt sortate crescător după coordonatele x ;
- nu vor exista două vapoare, sau un far și un vapor cu aceeași valoare pentru coordonata x ;
- pot exista două sau mai multe faruri cu aceeași coordonată x , caz în care ele vor fi unul lângă altul în fișierul de intrare (din moment ce sunt sortate după x);
- ordinea în care apar în fișierul de intrare farurile având aceeași coordonată x nu este definită;
- pot exista chiar două faruri identice.
- numerele reale din fișierul de intrare vor avea cel mult 4 zecimale exacte;
- rezultatul va fi verificat cu o precizie de 0,001 (rezultatul va fi considerat corect dacă modulul diferenței dintre rezultatul corect și cel furnizat de concurrent nu depășește 0,001);
- există întotdeauna soluție (pentru fiecare far i vor exista întotdeauna cel puțin Fn_i vapoare în stânga lui).

Exemplu

| sea.in | sea.out |
|----------|---------|
| 3 5 | 5.0990 |
| -0.5 0.5 | 0.7071 |
| -2 5 | 5.3852 |
| 3 4 | 0.7071 |
| -1 1 | 4.4721 |
| 0 1 | |
| 0 2 | |
| 0 1 | |
| 5 1 | |

Timpi de execuție:

- *Microsoft Windows*: 1,6 secunde/test
- *Linux*: 0,8 secunde/test

Câștigătorii

Clasa a V-a

Cătălin Bălan, Cluj - premiul I
Vlad Tătăranu, București - premiul I
Ioana Pandele, București - premiul III

Clasa a VI-a

Petre Clinciu-Glișca, Gorj - premiul I
Ada-Mihaela Solcan, București - premiul I
Otilia Stretcu, Mehedinți - premiul I

Clasa a VII-a

Liviu Mirea-Ghiban, București - premiul I
Alexandru-Petrișor Pajarcu, Dolj - premiul I

Clasa a VIII-a

Casian-Valentin Andrei, Argeș - premiul I
Dan Grigoriu, București - premiul II
Andrei Ștefan, București - premiul II

Membrii Lotului Național de Informatică:

Cristian Botău, Bihor
Mihai Ciucu, Dolj
Dan-Leonard Crestez, Brăila
Mircea Digulescu, București
Marius Dumitran, București
Constantin Dumitrașcu, Vrancea
Dan-Ionuț Fechet, Suceava

Silviu Gânceanu, București
Dan Ghinea, București
C^{tin}-Paul Grigoraș, Suceava
Emilian Miron, Bacău
Alexandru Moșoi, București
Marius Nicolae, Olt
Daniel Păsăilă, Suceava

Mircea Pașoi, Prahova
Ionut Prunache, Vrancea
Dan Spătărel, București
Sorin Stancu-Mara, București
George-Cristian Strat, Vaslui
Adrian Vladu, București