



ACM 2002

Pe data de 19 octombrie a avut loc la Universitatea Politehnica din București etapa regională pentru Europa de Sud-Est a concursului internațional de programare ACM. Vă prezentăm enunțurile celor opt probleme propuse spre rezolvare la acest concurs.

P080201: Bancomat

O bancă plănuiește să instaleze un nou automat bancar pentru retragere de numerar. Automatul este capabil să furnizeze bancnote ♣ (acest simbol reprezintă moneda folosită în țara în care este instalat bancomatul - poate reprezenta *dolari, euro, lire sterline* etc.) cu valoarea totală egală cu suma cerută de client. Bancomatul folosește exact N tipuri de bancnote cu valori diferite. Vom nota valorile celor N tipuri de bancnote prin D_k , $k = 1, \dots, N$. De asemenea, vom nota prin n_k rezerva (numărul) de bancnote cu valoarea D_k . De exemplu, am putea avea:

$N = 3$, $n_1 = 10$, $D_1 = 100$, $n_2 = 4$, $D_2 = 50$, $n_3 = 5$, $D_3 = 10$, ceea ce înseamnă că bancomatul are o rezervă de 10 bancnote cu valoarea de 100♣, 4 bancnote cu valoarea de 50♣ și 5 bancnote cu valoarea de 10♣.

Vom nota prin *cash* suma cerută de client. Va trebui să scrieți un program care determină cea mai mare sumă posibilă, mai mică sau egală cu *cash*, care poate fi retrasă, ținând cont de bancnotele din rezerva bancomatului.

Date de intrare

Datele de intrare vor fi citite din fișierul text **A.DAT**. Fiecare linie a acestui fișier reprezintă un set de date (corespunzător unei tranzacții) și are formatul:

cash N n₁ D₁ n₂ D₂ ... n_N D_N,
unde $0 \leq \text{cash} \leq 100000$ reprezintă suma cerută de client, $0 \leq N \leq 10$ reprezintă numărul tipurilor de bancnote, $0 \leq n_k \leq 1000$ reprezintă numărul bancnotelor disponibile care au valoarea D_k ($1 \leq D_k \leq 1000$, $k=1, \dots, N$). Numerele de pe o linie pot fi separate prin oricâte spații, iar datele de intrare sunt întotdeauna corecte.

Date de ieșire

Pentru fiecare set de date, rezultatul va fi afișat la ieșirea standard, fiecare răspuns fiind scris pe câte o linie.

Exemplu

A.DAT

```
735 3 4 125 6 5 3 350
633 4 500 30 6 100 1 5 0 1
735 0
0 3 10 100 10 50 10 10
```

Ieșire

```
735
630
0
0
```

Explicație

Primul set de date reprezintă o tranzacție în care suma cerută este 735♣. Bancomatul are o rezervă de 3 bancnote cu valoarea de 125♣, 6 bancnote cu valoarea de 5♣ și 3 bancnote cu valoarea de 350♣. În aceste condiții bancomatul va putea furniza suma exactă.

Pentru al doilea set de date bancomatul nu va putea furniza suma exactă, iar clientul va putea retrage suma de 630♣. Se observă că există mai multe posibilități de a obține această sumă folosind rezervele bancomatului.

Pentru al treilea set de date bancomatul nu are la dispoziție nici o bancnotă, deci nu este posibilă nici o extragere de numerar, iar pentru cel de-al patrulea set de date suma cerută este 0♣, deci bancomatul nu trebuie să folosească nici o bancnotă pentru a "elibera" această sumă.

P080202: Ținutul berii

Regele ACM 2002 a moștenit un teren și o sumă de bani. În testament se specifica clar că moștenitorul trebuie să construiască n orașe. De asemenea, trebuie să construiască străzi (acestea au două sensuri de circulație) între orașe, astfel încât un călător poate ajunge din fiecare oraș în oricare altul chiar dacă una dintre străzi este închisă pentru reparații. Toate străzile trebuie să fie construite în așa fel încât să poată fi privite ca fiind segmente de dreaptă. Mai mult, condiția ca un călător să poată ajunge dintr-un oraș în altul (chiar dacă o stradă este închisă), trebuie să fie satisfăcută în așa fel încât străzile să fie traversate integral. Nu există posibilitatea ca un călător să pornească pe o stradă, să se oprească la o intersecție aflată în afara orașelor și să continue pe o altă stradă.

Viitorii locuitori au decis să producă bere deoarece solul din această regiune permitea plantarea culturilor necesare. Regele a decis ca la fiecare intersecție de străzi (punctul de întâlnire a două sau mai multe străzi) să fie construit un stand cu bere. Așadar, dacă trei sau patru (sau zece) străzi se intersectează într-un singur punct, atunci există un singur loc în care poate fi amplasat standul corespunzător intersecției. Regele trebuie să aleagă pozițiile orașelor și străzile care vor uni orașe. Va trebui să îl sfătuiți cu privire la cea mai bună alegere, astfel încât să poată fi construite cât mai multe standuri cu bere.



Date de intrare

Prima linie a fișierului de intrare **B.DAT** conține numărul seturilor de date. Fiecărui set de date îi corespunde o linie care conține numărul n ($1 \leq n \leq 32767$) al orașelor care trebuie construite.

Date de ieșire

Pentru fiecare set de date va trebui să scrieți la ieșirea standard câte o linie care va conține numărul maxim de standuri cu bere care pot fi construite.

Exemplu

B.DAT	Ieșire
3	0
3	1
4	5
5	

P080203: Iaht

Sunteți fericitul proprietar al unui iaht foarte frumos. Aveți multe cereri de închiriere a iahtului pentru perioada verii și ați hotărât să obțineți un profit cât mai mare. Pentru fiecare client cunoașteți numărul de zile pentru care dorește să închirieze iahtul și o listă de opțiuni. O opțiune constă dintr-o sumă de bani și un termen limită (numărul maxim de zile dintre începutul verii și ziua în care s-ar termina perioada de închiriere pentru clientul respectiv). Veți primi suma corespunzătoare unui termen limită numai dacă reușiți să închiriați iahtul astfel încât călătoria să se încheie cel târziu în ziua corespunzătoare termenului limită.

De exemplu, prietenul vostru *Jack* vrea să facă o călătorie de 20 de zile prin *Marea Mediterană*. Opțiunile pe care vi le pune la dispoziție sunt:

- Dacă termenul limită este de 60 de zile, primiți 1000 de EURO; veți primi această sumă numai dacă durata de închiriere durează exact 20 de zile și începe cel târziu în a 41-a zi ($60 - 20 + 1$) de la începutul verii;
- Dacă termenul este de 70 de zile primiți 800 de EURO.

Dacă un client nu contribuie la maximizarea profitului, puteți renunța la clientul respectiv. Îi cunoașteți pe toți clienții și știți care sunt opțiunile fiecăruia. Va trebui să închiriați iahtul în ordinea în care sosesc cererile clienților.

Scrieți un program care, dându-se opțiunile clienților, determină profitul maxim care poate fi obținut.

Date de intrare

Datele de intrare vor fi citite din fișierul text **C.DAT**, fiecare set de date din fișier având următorul format:

- pe prima linie valoarea n - numărul clienților ($n \leq 100$);
- pe n linii distincte numărul de zile (cel mult 100) pentru care un client dorește să închirieze iahtul;
- pe următoarea linie numărul total al opțiunilor (pentru toți clienții);
- pe următoarele linii câte o opțiune având formatul *client*, *termen*, *sumă* unde *client* este un număr din intervalul $[1, n]$, iar *termen* este o valoare mai mică decât 100.

Seturile de date din fișier vor fi separate printr-o linie vidă.

Date de ieșire

Pentru fiecare set de date va trebui să scrieți la ieșirea standard câte o linie pe care se va afla profitul maxim care poate fi obținut. Datele de ieșire corespunzătoare unor seturi de date consecutive vor fi separate printr-o linie vidă.

Exemplu

C.DAT	Ieșire
3	26
2	
2	
4	
4	
1 2 14	
3 4 25	
2 4 12	
3 3 10	

P080204: Galerie de artă

Galeriile de artă din noua clădire a *Centrului pentru Cooperare Balcanică* au formă poligonală (nu neapărat convexă).

Când are loc o expoziție de anvergură, apare problema supravegherii tuturor tablourilor expuse. Va trebui să scrieți un program care, pentru o galerie dată, determină aria suprafeței din care pot fi observate toate punctele de pereții galeriei.

În figura 1 este prezentată o hartă a unei galerii, iar în figura 2 este hașurată suprafața cerută.

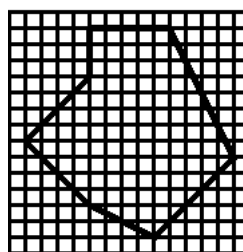


Figura 1

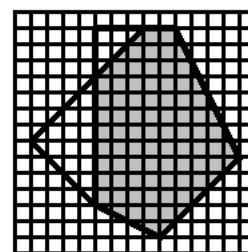


Figura 2

Date de intrare

Prima linie a fișierului de intrare **D.DAT** conține numărul T al seturilor de date. Fiecare set de date conține o linie pe care se află numărul N ($5 \leq N \leq 1500$) al vârfurilor poligonului și N linii care conțin câte două numere întregi care reprezintă coordonatele unuia dintre vârfuri. Coordonatele pot fi reprezentate ca numere întregi pe 16 biți și sunt separate printr-un singur spațiu.

Date de ieșire

Pentru fiecare set de date va trebui să scrieți la ieșirea standard câte o linie pe care se va afla aria suprafeței cerute (valoarea va fi scrisă folosind două zecimale exacte).



Exemplu

D.DAT	Ieșire
1	80.00
7	
0 0	
4 4	
4 7	
9 7	
13 -1	
8 -6	
4 -4	

P080205: Joc

Recent a avut loc un campionat internațional de volei. Suntem interesați de probabilitatea de a câștiga partida pe care o are fiecare dintre cele două echipe în condițiile în care avem la dispoziție anumite informații despre echipe. Vom prezenta regulile jocului și informațiile necesare:

- Partida se desfășoară între două echipe A și B .
- Prima echipă care câștigă K seturi, câștigă partida.
- Un set constă în mai multe runde; fiecare rundă este câștigată de o echipă; pentru fiecare rundă câștigată echipa primește un punct.
- Setul este câștigat de echipa care acumulează L puncte.
- Dacă echipa A se află la serviciu, atunci există o probabilitate de $Pa\%$ să câștige runda și o probabilitate de $(100 - Pa)\%$ să o piardă.
- Dacă echipa B se află la serviciu, atunci există o probabilitate de $Pb\%$ să câștige runda și o probabilitate de $(100 - Pb)\%$ să o piardă.
- Dacă o rundă nu este prima dintr-un set, atunci servește echipa care a câștigat runda precedentă.
- Dacă un set nu este primul dintr-o partidă, atunci la prima rundă a setului respectiv va servi echipa care nu a servit la prima rundă a setului anterior.
- Cele două echipe au șanse egale de a servi în prima rundă a primului set.

Așadar, va trebui să determinați probabilitatea, exprimată în procente, ca echipa A să câștige partida.

Date de intrare

Prima linie a fișierului de intrare **E.DAT** conține numărul seturilor de date. Pentru fiecare set de date fișierul de intrare va conține o linie pe care se vor afla numerele întregi Pa , Pb , K și L ($1 \leq K \leq 100$, $1 \leq L \leq 100$).

Date de ieșire

Pentru fiecare set de date va trebui să scrieți la ieșirea standard câte o linie care va conține un număr (scris cu o zecimală exactă), reprezentând șansele de câștig ale echipei A .

Exemplu

E.DAT	Ieșire
2	93.8
100 50 1 3	99.5
100 1 1 1	

P080206: Proces server

Una dintre paradigmele procesării paralele este paradigma producător / consumator. Ea poate fi implementată folosind un proces server și mai multe procese client. Procesele client pot reprezenta producători sau consumatori. Procesul server păstrează informații referitoare la procesele client. Fiecare proces este identificat printr-un cost care este un număr cuprins între 1 și 10000. Nu pot exista mai mult de 10000 de procese care să aibă același cost. Coada este implementată în așa fel încât să poată satisface trei tipuri de cereri:

- ax - se adaugă în coadă un proces cu costul x ;
- r - un proces este eliminat din coadă (dacă este posibil) în concordanță cu politica de eliminare actuală;
- pi - stabilește politica i pentru server, unde i poate avea una dintre valorile 1 și 2. Politica implicită este 1;
- e - indică sfârșitul listei de cereri.

Există două politici de eliminare:

- 1 - se elimină procesul cu cel mai mic cost;
- 2 - se elimină procesul cu cel mai mare cost.

Serverul va afișa costul unui proces eliminat numai dacă numărul de ordine al procesului eliminat se află în lista de eliminare.

Va trebui să scrieți un program care simulează procesul server.

Date de intrare

Datele de intrare vor fi citite din fișierul text **F.DAT**. Fiecare set de date din acest fișier va avea următorul format:

- costul maxim al proceselor;
- lungimea listei de eliminare;
- lista de eliminare - o listă care conține numere de ordine ale proceselor care vor fi afișate; de exemplu, lista 1 4 indică faptul că vor fi scrise costurile primului și celui de-al patrulea proces eliminat;
- lista cererilor care conține câte o cerere pe fiecare linie.

Fiecare set de date se încheie printr-o cerere e , iar seturile de date sunt separate prin linii vide.

Date de ieșire

Programul vostru va scrie la ieșirea standard costul fiecărui proces eliminat dacă numărul de ordine al cererii de eliminare se află în lista de eliminare și coada nu este vidă în momentul respectiv (dacă este vidă se va scrie valoarea -1). Fiecare cost va fi scris pe câte o linie, iar costurile corespunzătoare unor seturi de date consecutive vor fi separate printr-o linie vidă.

Exemplu

F.DAT	Ieșire
5	2
2	5
1 3	
a 2	
a 3	
r	



a 4
p 2
r
a 5
r
e

P080207: Rețea

Vi s-a dat sarcina să proiectați o rețea care să lege anumite puncte dintr-o zonă extinsă. Se dau punctele din zonă și o mulțime de trasee posibile pentru cablurile care pot conecta perechi de puncte. Pentru fiecare traseu posibil dintre două puncte se cunoaște lungimea cablului necesar conectării celor două puncte folosind traseul respectiv. Între două puncte pot exista mai multe trasee. Se presupune că folosind traseele date se pot interconecta (direct sau indirect) oricare două puncte.

Va trebui să proiectați rețeaua astfel încât să existe o conexiune (directă sau indirectă) între oricare două puncte ale rețelei (toate punctele sunt interconectate, dar nu neapărat prin cabluri directe) și costul total al cablurilor folosite este minim.

Date de intrare

Fișierul de intrare **G.DAT** conține mai multe seturi de date, fiecare dintre ele caracterizând o zonă. Prima linie dintr-un set de date conține numărul P al punctelor și numărul R al traseelor posibile. Următoarele R linii conțin câte trei numere; primele două identifică cele două puncte unite prin traseu, iar al treilea reprezintă lungimea traseului. Numerele sunt separate prin spații. Ultimul set de date conține doar valoarea $P = 0$.

Există cel mult 50 de puncte, iar lungimea maximă a unui traseu este 100. Numărul traseelor posibile este nelimitat. Punctele sunt identificate prin numere întregi cuprinse între 1 și P . Traseele dintre două puncte i și j pot fi date fie prin $i j$, fie prin $j i$.

Date de ieșire

Pentru fiecare set de date va trebui să scrieți la ieșirea standard câte o linie care va conține lungimea totală a cablurilor folosite pentru interconectare.

Exemplu

G.DAT	Ieșire
1 0	0
	17
2 3	16
1 2 37	26
2 1 17	
1 2 68	
3 7	
1 2 19	
2 3 11	
3 1 7	

1 3 5
2 3 89
3 1 91
1 2 32

5 7
1 2 5
2 3 7
2 4 8
4 5 11
3 5 10
1 5 6
4 2 12

0

P080208: Număr șmecher

Vom numi număr șmecher un șir A care conține N numere întregi din mulțimea $\{0,1,2\}$ (de exemplu, $A_0 = 1, A_1 = 1, A_2 = 0$ și $A_3 = 2$). Un număr șmecher este numit **UNU** dacă $A_0 = 1$ și $A_I = 0$ pentru $I = 1, 2, \dots, N - 1$.

Considerăm următoarea operație dintre două numere șmechere, numită înmulțire stelată:

$$C_k = \sum_{i=0}^k A_i \cdot B_{k-i} + \sum_{i=k+1}^{N-1} A_i \cdot B_{N+k-i}.$$

Rezultatul operației este un șir C care nu este neapărat un număr șmecher. Această operație va fi reprezentată prin simbolul $*$. Mai mult, avem la dispoziție și o operație modulo care poate fi aplicată rezultatului unei înmulțiri stelate: $(C \bmod Q)_i = C_i \bmod Q$, unde Q este un număr întreg pozitiv.

Se dă un număr șmecher A și o valoare Q . Va trebui să determinați numărul șmecher invers B . Acesta satisface relația $(A * B) \bmod Q = \text{UNU}$.

Date de intrare

Prima linie a fișierului de intrare **H.DAT** conține numărul K al seturilor de date. Fișierul conține două linii pentru fiecare test. Prima dintre acestea conține numerele întregi N ($5 \leq N \leq 50$) și Q ($2 \leq Q \leq 100$), separate prin spații. Cea de-a doua linie conține N numere întregi din mulțimea $\{0, 1, 2\}$ care reprezintă numărul șmecher A .

Date de ieșire

Pentru fiecare dintre cele K seturi de date se va scrie câte o linie la ieșirea standard. Dacă există un număr șmecher invers, linia va conține mesajul "A solution can be found", iar în caz contrar se va scrie "No solution".

Exemplu

H.DAT	Ieșire
2	A solution can be found
2 5	No solution
1 0 1 0 1	
65 8	
1 2 2 2 1 1 2 2	