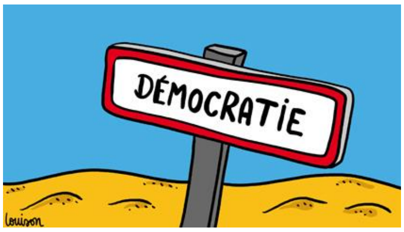
Concurs online de informatică
Categoria PROGRAMARE

Secţiunea 7-8 avansaţi

PROBLEMA 1 **100 puncte**

DEMOCRATIE



Arpsod are în curtea sa N copaci foarte bătrâni, așezați în linie și numerotați de la 1 la N. Fiecare copac are o înălțime cunoscută, Hi. Există riscul ca la un vânt mai puternic aceştia să cadă, provocând stricăciuni. Astfel Arpsod a angajat doi muncitori pentru a-i tăia copacii. Primul muncitor va începe să taie copacii în ordinea 1, 2, 3, ... ,N iar cel de-al doilea în ordinea N, N-1, N-2, ... 1. Fiind un tărâm democratic, fiecare muncitor doreşte să fie plătit pentru fiecare metru pe care îl taie. Muncitorul 1 are un tarif de T1 pe metru iar muncitorul 2 un tarif de T2 pe metru. Dacă un muncitor a început să taie un copac, acesta îl va tăia integral. Din motive de protecţie a muncii, muncitorilor nu le este permis să lucreze simultan. De aici apare următoarea pretenţie: dacă după tăierea unui copac, muncitorul nu este înlocuit de colegul său, acesta va cere un cost suplimentar C pentru a rămâne să taie în continuare.

De exemplu, dacă avem 3 copaci: 1, 2, 3 și muncitorul 1 taie singur toți copacii, acesta va cere un cost suplimentar de 2 ori (pentru copacul 2 și copacul 3)

Cerința
Arpsod vă cere să determinați costul minim pe care îl poate plăti astfel încât toți cei N copaci să fie tăiați.

Date de intrare
Pe prima linie a fişierului ***democratie.in*** se va afla numărul natural N, reprezentând numărul de copaci. Pe cea de-a doua linie vor exista N numere naturale nenule reprezentând înălțimile celor N copaci.
Pe cea de-a treia linie se vor afla două numere naturale T1 și T2 reprezentând tariful pe metru al muncitorului 1 respectiv al muncitorului 2.
Pe ultima linie se vor afla două numere naturale C1 și C2 reprezentând costul suplimentar cerut de muncitorul 1 respectiv muncitorul 2.

Date de ieșire
În fişierul ***democratie.out*** se va scrie, pe prima şi singura linie din fişier, costul minim pe care Arpsod trebuie să-l plătească.

Restricţii şi precizări:
- $1 \leq N \leq 100.000$
- $1 \leq T1, T2 \leq 100$
- $1 \leq C1, C2 \leq 10.000$
- $1 \leq Hi \leq 100$
- Se garantează că pentru 20% din teste $1 \leq N \leq 10$
- Costul suplimentar este același indiferent de înălțimea copacului ce va fi tăiat.
- Este posibil ca un muncitor să taie singur toți copacii.
- Un muncitor va tăia complet un copac.
- Cam scumpă democrația asta!

Ploieşti, 28 mai 2016

Exemplu

democratie.in	democratie.out	Explicaţie
4 1 2 3 4 7 2 3 9	34	Ordinea muncitorilor: M2 -> M1 -> M2 -> M2 Costul: (2*4) + (7*1) + (2*3) + (2*2 + 9)

Timp maxim de execuţie: 0.5 secunde/test
Memorie totală disponibilă: 4MB şi 2MB pentru stivă
Dimensiunea maximă a sursei: 5Kb

PROBLEMA 2 **100 puncte**

TREASURE HUNT

Alex s-a decis să organizeze pentru colegii lui un concurs de orientare turistică, pe care l-a intitulat "Treasure Hunt", nu pentru că ar fi avut ceva bogății de ascuns, ci pentru a-i face curioși și a-i mobiliza să mai lase puțin joaca pe calculator și să facă mișcare în aer liber. Pentru aceasta el a cercetat terenul pe care s-a decis să organizeze acest concurs și a identificat n puncte posibile de amplasare a posturilor de control, prin care concurenții să treacă obligatoriu și de unde să primească indicii referitoare la următorul punct. Bineînțeles că în punctele de control există și "comori" ascunse, care au asociate anumite punctaje, de valori cunoscute. A notat pe o hartă coordonatele (x,y) ale acestor puncte, în ordinea necesară parcurgerii lor, dar și altitudinea la care se află acestea și punctajul p atribuit "comorii" din acel punct. Problema a apărut mai târziu, atunci când Alex s-a decis să nu lase un concurent să se oprească în toate punctele, pentru că atunci colegii lui ar găsi un motiv să mai tragă de timp pentru a se odihni și concursul ar dura prea mult. Așa că a stabilit să permită maxim M opriri din cele N puncte, cu condiția ca între două opriri succesive distanța de pe traseu să nu fie mai mică decât o valoare impusă, d, stabilită de Alex printr-o metodă proprie. Trecerea prin toate punctele este obligatorie, așa că distanța se calculează ca suma distanțelor dintre puncte. Curios din fire, Alex vrea să știe:

1. Care este efortul total pentru parcurgerea întregului traseu și care este distanța maximă între două puncte de pe traseu. Efortul trebuie calculat după o formulă pe care tot Alex a stabilit-o ca suma eforturilor de pe fiecare tronson în parte, definit astfel
$$e=\begin{cases} d + d * \Delta h/10, & dacă\ urcă \\ d + d * \frac{|\Delta h|}{5*10}, & dacă\ coboară \end{cases}$$
, unde Δh este diferența de altitudine dintre două puncte consecutive de pe traseu, chiar dacă în acestea concurentul nu se oprește.

2. Care este punctajul maxim pe care îl poate obține un concurent, și care ar trebui să fie punctele de control în care să se oprească pentru a le obține.

Din păcate Alex nu e prea bun la informatică, așa ca vă roagă pe voi să-l ajutați.

Date de intrare

Fișierul treasure.in și are următoarea formă:
- Pe prima linie un jnumăr natural z=1 sau z=2
- Pe următoarea linie numerele N, M și d, reprezentând numărul de puncte de control stabilite, numărul maxim de opriri acceptate, respectiv distanța minimă între două opriri succesive.

- Pe următoarele N linii câte 4 numere x,y,h,p reprezentând coordonatele, altitudinea și punctajul comorii din fiecare punct. Punctele 1 și n vor rămâne obligatoriu pe traseu și nu conțin comori, așadar nu li se impune nici o condiție dintre cele stabilite mai sus (nu se numără în cele M opriri, iar distanța față de punctele unei opriri alese poate fi mai mica decât d).

Date de ieșire

Fișierul treasure.out are forma:

Dacă z=1, pe prima linie un număr real care reprezintă distanța maximă între două puncte de pe traseul inițial și pe a doua linie un alt număr real care reprezintă efortul total pentru parcurgerea întregului traseu

Dacă z=2, pe prima linie punctajul maxim care poate fi obținut cu cel mult M opriri și pe următoarea linie numerele de ordine ale punctelor de pe traseu unde se fac opririle corespunzătoare acestui caz.

Restricții și precizări

1 ≤M≤ N ≤ 10000, 0≤ p ≤ 100, 0<=d<=1000

Coordonatele punctelor și altitudinea sunt numere întregi cu maxim 4 cifre, distanțele și efortul sunt numere reale calculate cu exact 2 cifre, fără aproximare.

Exemple

treasure.in	**treasure.out**	**Explicaţii**
1 10 5 3 0 0 0 0 2 0 3 4 3 0 3 7 4 0 5 10 5 0 5 5 6 0 3 9 7 0 4 10 8 0 4 15 10 0 6 15 15 0 0 0	5 16.94	Între punctele 1 și 2 distanța este 2 și urcă cu 3, deci efortul va fi 2+2*3/10=2.6 Între punctele 2 și 3 distanța este 1 și Δh =0, deci efortul va fi 1 Între punctele 3 și 4 distanța este 1 și Δh =2, deci efortul va fi 1,2 Între punctele 4 și 5 distanța este 1 și Δh =0, deci efortul va fi 1, etc rezultă efortul total 16.94 Distanta maxima este 5 intre punctele 9 și 10
2 10 5 3 0 0 0 0 2 0 3 4 3 0 3 7 4 0 5 10 5 0 5 5 6 0 3 9 7 0 4 10 8 0 4 15 10 0 6 15 15 0 0 0	35 1 4 7 9 10	Punctajul maxim care se poate obține este 35, când concurenții se opresc în punctele 1,4,7,9 și 10 din care obțin punctajele 0+10+10+15+0

Timp maxim de execuție: **0,2** secunde/test.
Memorie totală disponibilă: 2 MB, din care **2** MB pentru stivă
Dimensiunea maximă a sursei: **5** KB.

PROBLEMA 3 **100 puncte**

BOMBOANE

Se consideră un şir de **N** numere ce reprezintă numărul de bomboane din **N** punguțe așezate în linie. Georgel porneşte din dreptul primei punguțe şi ia din ea o bomboană, apoi trece la următoarea şi ia din ea două bomboane şi continuă în acest fel luând de fiecare data cu o bomboană mai mult decât a luat din punga precedentă. Dacă au mai rămas bomboane în punguțe astfel încât el să mai poată face o tură completă, continuă să ia bomboane, câte una în plus față de punguța din care a luat anterior. Dacă nu mai sunt bomboane, se opreşte şi își numără bomboanele adunate în total.

Cerinţe
Cunoscându-se numărul **N** de punguțe și numărul de bomboane din fiecare punguță, să se stabilească numărul de bomboane pe care le va aduna Georgel.

Date de intrare
Din fișierul **bomboane.in** se citesc, de pe prima linie numărul **N** de punguțe, iar de pe linia a doua, **N** numere naturale reprezentând numărul de bomboane din fiecare punguță, în ordinea în care se află. Numerele sunt despărțite între ele prin spații.

Date de ieşire
În fişierul **bomboane.out** se afişează valoarea **M**, reprezentând numărul de bomboane pe care le poate aduna Georgel.

Restricţii
- $1 \leq N \leq 100000$
- Numărul bomboane din fiecare punguță este un număr natural nenul mai mic decât 10^{12}.

Exemplu

bomboane.in	bomboane.out	Explicații
6 **180 90 141 55 63 120**	**300**	Georgel pornește de la punguța numărul 1 și ia o bomboană, din a doua ia două bomboane, din a treia 3 bomboane, din a patra 4, din a cincea 5, din a șasea 6; la următoarea tură ia din prima 7 bomboane, din a doua 8, din a treia 9, din a patra 10 și în continuare 11, 12; la următoarea tura ia 13, 14, 15, 16 17, 18, iar la a patra tură ia 19, 20, 21, 22, 23, 24 bomboane. La a cincea tură se oprește deoarece în unele punguțe nu se mai află numărul de bomboane necesar pentru a termina o nouă tură completă. Astfel adună în total **300** de bomboane.

Timp maxim de execuţie: 0,2 secunde / test
Memorie totală disponibilă: 8MB şi 2MB pentru stivă
Dimensiunea maximă a sursei: 5Kb

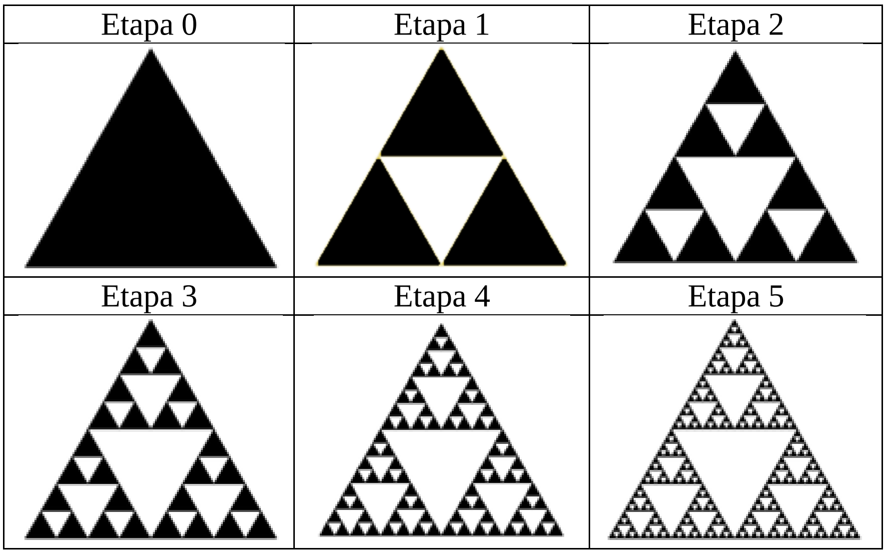
PROBLEMA 4 **100 puncte**

MORUM

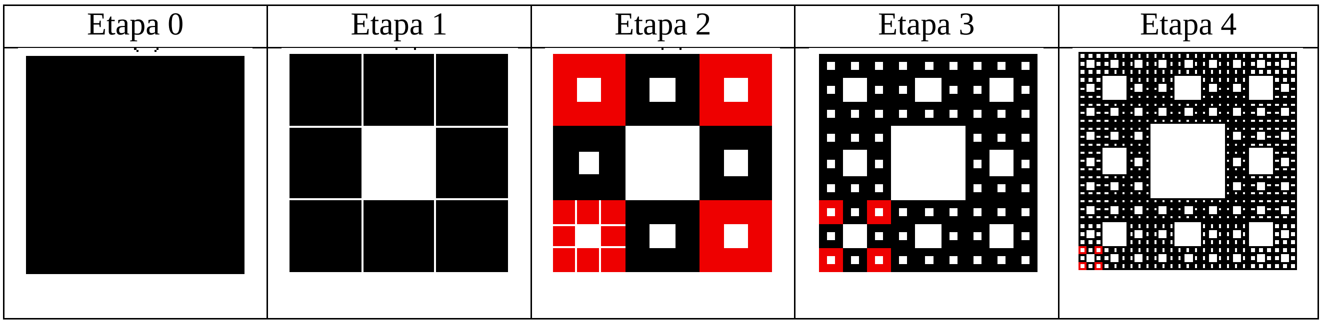
Magazinele IKEA au spre vânzare o categorie aparte de covoare, MORUM, pentru uz intern sau extern, rezistente la ploaie, soare, zăpadă şi noroi. Producătorul poate construi covoare MORUM clasice, dar poate construi şi modele la alegerea clientului. Sunt **trei** modele puse la dispoziție de producători, generate în mai multe etape ale procesului de fabricație. Un covor se generează în mai multe etape, la fiecare etapă se decupează covorul conform desenelor de mai jos.

De exemplu, dacă covorul are formă de triunghi, în prima etapă se decupează un triunghi reprezentat în desen prin zona albă. În a doua etapă, din fiecare zonă triunghiulară neagră se va decupa folosind același procedeu ca la prima etapă un alt triunghi.

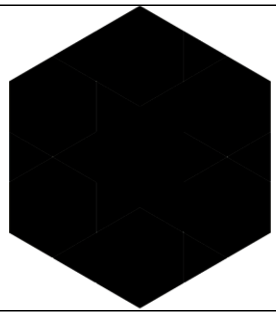
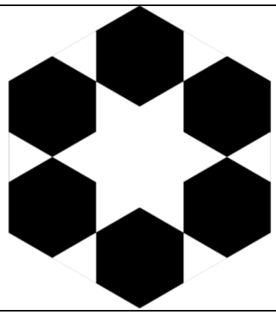
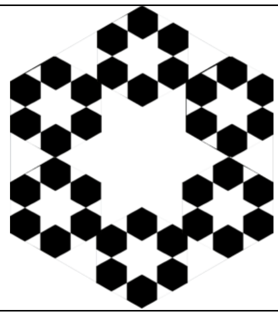
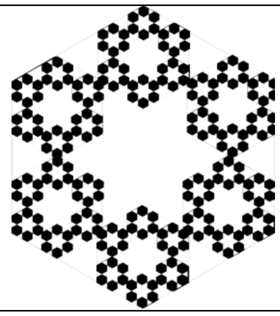
Model 1



Model 2



Model 3

Etapa 0	Etapa 1	Etapa 2	Etapa 3
			

Deoarece, în procesul de dantelare, covoarele vor deveni din ce în ce mai fragile, producătorul solicită fiecărui cumpărător procentul din suprafața inițială care poate fi îndepărtat.

Cerință
Cunoscând latura **l** a covorului, modelul **m** ales, procentul **p** din suprafața inițială a covorului care poate fi decupat, determinați gradul de dantelare al covorului (etapa până la care se poate proceda la modelare), precum și lungimea conturului și suprafața covorului după decupare (perimetrul și aria se vor afişa fiecare prin câte o fracție).

Date de intrare
Fişierul ***morum.in*** conține pe prima linie, separate prin spațiu, valorile **l**, **m** şi **p**.

Date de ieşire
Fişierul ***morum.out*** va conține pe prima linie gradul de dantelare, iar pe a doua linie un număr natural reprezentând numărătorul fracției ce reprezintă suprafața conturului, iar pe a treia linie numitorul fracției ce reprezintă suprafața suprafata conturului. Pe linia patru se afişează un număr natural ce reprezintă numărătorul fracției ce reține lungimea covorului, iar pe a cincea linie numitorul acesteia.

Exemplu

morum.in	**morum.out**	**Explicație**
81 2 34	3 3359232 729 40176 27	Pentru a construi covorul se folosesc 3 etape de dantelare. Fracția ce reprezintă suprafața covorului după decupare este $\frac{3359232}{729}$=4608. Fracția ce reprezintă perimetrul covorului după decupare $\frac{40176}{27}$=1488.
64 1 50	2 6377472 6400 1728 4	Pentru a construi covorul se folosesc 2 etape de dantelare. Fracția ce reprezintă suprafața covorului după decupare este $\frac{6377472}{6400}$=996.48. Fracția ce reprezintă perimetrul covorului după decupare $\frac{1728}{4}$=432.
81 3 85	4 54482544 16200 29856	Pentru a construi covorul se folosesc 4 etape de dantelare. Fracția ce reprezintă suprafața covorului după decupare este $\frac{54482544}{16200}=$ 3363,12

	81	Fracția ce reprezintă perimetrul covorului după decupare $\frac{29856}{81}$=7776.

Restricții și precizări
- latura covorului este un număr natural $l \leq 1000000000$;
- procentul p este un număr natural cuprins între 1 și 99;
- toate poligoanele inițiale și cele obținute prin decupare sunt regulate;
- gradul de dantelare este un număr natural, reprezentând etapa la care se oprește procesul de decupare;
- în calculele se va considera că $\sqrt{3} = 1.73 = \frac{173}{100}$;
- lungimea conturului şi aria vor fi reprezentate prin fracții; fracţiile nu sunt unice.
- în urma dantelării materialul nu se destramă și formele din ce în ce mai mici care rămân vor rămâne prinse de restul covorului

Timp maxim de execuție: 0,5 secunde / test
Memorie totală disponibilă: 2MB şi 1 MB pentru stivă
Dimensiunea maximă a sursei: 10 Kb