



Concursul național ”StartIT”, ediția a IX-a, 25.XI.2017
Clasele VII– VIII

Problema PLATA

100 puncte

Eroul nostru, Costy merge la magazin pentru a-și cumpăra biscuiți. Vânzătorul îi spune că biscuiții costă S nasturi, și că dorește ca plata lor să fie făcută cu n tipuri diferite de nasturi. De asemenea, vânzătorul precizează că ar dori ca numărul de nasturi din fiecare tip i să depășească valoarea $x[i]$, dar, să nu depășească valoarea $y[i]$. Presupunând că băiatul are o infinitate de nasturi din fiecare tip și că dorește să rămână cu cât mai mulți nasturi de tipurile $n, n-1, n-2, \dots$ în buzunar, ajutați-l să efectueze plata și să pună cât mai repede mâna pe biscuiți.

Cerință

Scrieți un program care să determine o modalitate de plată a sumei, respectând condițiile din enunț.

Date de intrare

Fișierul de intrare *plata.in* conține: Pe prima linie, $N S$, reprezentând numărul de tipuri de nasturi și suma care trebuie plătită. Pe următoarele N linii, două numere separate printr-un spațiu, $x[i] y[i]$, reprezentând numărul minim și numărul maxim de nasturi de tipul i , care pot fi aleși.

Date de ieșire

Fișierul de ieșire *plata.out* va conține o linie cu n numere, separate prin câte un spațiu, $r[1] r[2] r[3] \dots r[n]$ reprezentând numărul de nasturi de tipul i ales, sau cuvântul « *imposibil* » (fără ghilimele) dacă plata nu se poate realiza în condițiile date

Restricții și precizări

- $1 \leq n \leq 1000$
- $1 \leq x \leq y \leq 1000$
- $1 \leq s \leq 1000000$

Nasturii au aceeași valoare, indiferent de tipul acestora.

Exemplu

plata.in	plata.out
5 18 1 2 4 6 1 4 5 8 1 3	2 6 4 5 1

Explicație:

Eroul nostru va face plata în felul următor:

2 nasturi de tipul 1, 6 nasturi de tipul 2, 4 nasturi de tipul 3, 5 nasturi de tipul 4, 1 nasture de tipul 5,
 În total $2 + 6 + 4 + 5 + 1 = 18$ nasturi

De observat că dacă se alegea soluția : 1 6 4 6 1, aceasta nu era corectă deoarece Costy trebuia să scoată din buzunar un nasture de tip mai mare în schimbul celui de tip 1.



Problema - BONA

100 puncte

Enunț:

În camera copiilor problema spațiului este esențială. De aceea locul unde sunt depozitate jucăriile este asemănător unei matrici pătratice, dispuse vertical, fiecare jucărie ocupând un locație bine stabilită.

Entuziasmul și bucuria copiilor în a se juca este bine cunoscută, dar și "disponibilitatea" acestora de a ordona (reașeza) jucăriile la locul prestabilit este proverbială. Cum, după o rundă de joacă, întotdeauna urmează un episod din serialul de desene animate preferat, copii așează la întâmplare jucăriile.

Cunoscând numărul M de jucării, iar pentru fiecare jucărie locația în care copii au pus jucăria (linie, coloană), precum și locația unde trebuie corect pusă aceasta (linie, coloană), ajutați bona să reasăze jucăriile astfel încât numărul de mutări să fie minim. În cazul în care locația unde trebuie mutată o jucărie este ocupată, atunci bona depozitează temporar jucăria într-o locație specială, dacă este liberă, până când locația unde trebuie mutată se va elibera.

Cerințe:

Știind locația pentru fiecare jucărie determinați:

- numărul minim de mutări ce nu necesită folosirea locației speciale
- numărul minim de mutări necesar rearanjării tuturor jucăriilor

Date de intrare:

Fișierul de intrare *bona.in* conține pe prima linie două numere naturale N, M ce reprezintă în ordine, numărul de linii și coloane ale dulapului/matrice folosit pentru jucării, respectiv numărul de jucării. Pe următoarele M linii un set de patru numere naturale ce reprezintă locația inițială (linia, coloana), respectiv locația finală (linia, coloana) pentru fiecare jucărie.

Date de ieșire:

Fișierul de ieșire *bona.out* va conține pe prima linie o valoare ce reprezintă numărul minim de mutări ce nu necesită folosirea locației speciale, iar pe a doua linie numărul minim de mutări necesar rearanjării tuturor jucăriilor.

Restricții și precizări:

- $1 \leq N \leq 1000$
- $1 \leq M \leq 10000$
- a) locațiile inițiale sunt distincte între ele, locațiile finale sunt distincte între ele
- b) două jucării nu pot ocupa în același timp aceeași locație.
- c) rezolvarea primei cerințe asigură 30% din punctajul unui test
- d) pentru datele de intrare se asigură existența unei soluții

Exemplu:

bona.in	bona.out
4 5	3
1 1 3 4	7
3 4 1 1	
3 1 4 3	
1 2 2 4	
2 4 1 2	



Explicație:

Jucăria 1 trebuie mutată din locația (1,1) în locația (3,4), dar aceasta este ocupată. Bona mută jucăria în locația specială. Jucăria 2 se mută din (3,4) în (1,1), eliberând locația (3,4). Jucăria 1 se mută din locația specială în locația finală (3,4).

Jucăria 3 se mută din (3,1) în (4,3). Jucăria 4 trebuie mutată din locația (1,2) în locația (2,4), dar aceasta este ocupată. Jucăria 4 este mutată de bonă în locația specială.

Jucăria 5 se mută din (2,4) în (1,2). Jucăria 4 se mută din locația specială în locația finală (2,4).