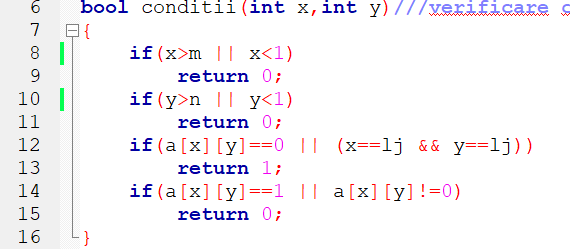
PROBLEMA 8

Variabilele problemei sunt:

* a[100][100]-matricea
* i , j-variabilele de parcurgere
* n-nr de coloane
* m-nr de linii
* lr , cr-linia,respectiv coloana din matrice ale lui Romeo
* lj,cj- linia,respectiv coloana din matrice ale Juiletei

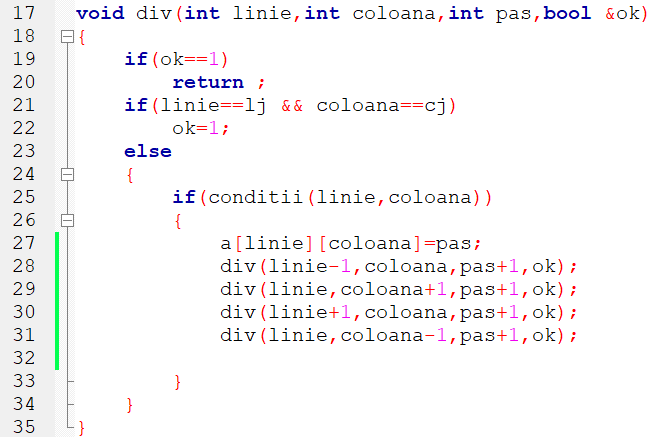
Rezolvarea problemei este efectuata cu 3 functii:

* Funcția “conditii”:



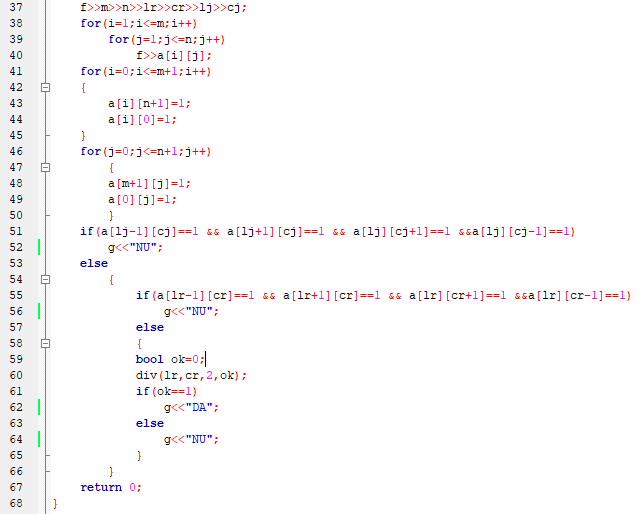
Functia are ca parametrii 2 variabile de tip int ,x si y, variabile pe care vom verifica urmatoarele conditii :

* + x>m (verifica daca elemental a depasit numarul de linii in partea de jos) && x<1(verifica daca elemental a depasit numarul de linii in partea de sus) va returna 0 daca este adevarat deoarece este un caz nefavorabil.
  + y>n(verifica daca elemental a depasit numarul de coloane in partea dreapta) && y<1((verifica daca elemental a depasit numarul de coloane in partea stanga) idem primei conditii.
  + a[x][y]==0(daca urmatorul element este egal cu 0,inseamna ca l putem parcurge,adica nu este zid) || x==lj && y==lj (daca coordonatele actuale sunt egela cu cele ale Julietei inseamna ca am rezolvat problema).
  + a[x][y]==1(caz nefavorabil deoarece ne lovim de un zid) || a[x][y]!=0(acel element este deja parcurs)
* Functia „div” :



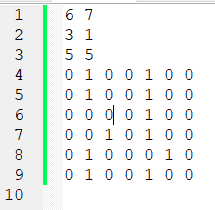
Functia are ca parametrii linia respectiv coloana curenta, variabila pas care reprezinta pasul curent la care ne aflam, variabila ok care ne spune daca am ajuns la Julieta sau nu.

* Daca ok=1 programul se va intoarce in recursivitate(daca este cazul) si va oprii executia tuturor apelarilor trecute.
* Daca linia este egala cu linia julietei si coloana este egala cu coloana Julietei ok=1 adica am ajuns la Julieta.
* Altfel,verificam conditiile pozitiei curente sa vedem daca suntem intr un zid (a[linie][coloana]=1) sau daca putem parcurge elementul curent pentru a verifica urmatoarele pozitii(sus,jos,stanga dreapta).
* a[linie][coloana]=pas pentru a stii daca acel element a fost deja parcurs sau nu.
* Pentru a imparii matricea pe mai multe drumuri apelam recursiv:
  + div(linie-1,coloana,pas+1,ok) ne va duce la elementul de sus.
  + div(linie,coloana+1,pas+1,ok) ne va duce la elementul din dreapta.
  + div(linie+1,coloana,pas+1,ok) ne va duce la elementul de jos
  + div(linie,coloana-1,pas+1,ok) ne va duce la elemntul din stanga.
* Functia „main” :



* + Vom citi n,m,lr,cr,lj,cj din fisier,apoi matricea a.
  + Initializam elementele exterioare matricii cu 1 pentru a creea un perete exterior in stanga,dreapta,sus,jos.
  + Verificam daca Julieta este prinsa intre pereti si vom afisa „NU”,pentru ca in acest caz este imposibil sa ajungem la Julieta.
  + Verificam daca Romeo este prins intre pereti si vom afisa „NU”,pentru ca in acest caz este imposibil sa ajungem la Julieta.
  + Altfel apelam functia „div” si daca ok=1 afisam „DA” altfel „NU”.

In fisierul „date.in” vom citii urmatoarele numere:



Iar in date.out va afisa urmatorul mesaj.



Programul are o complexitate O(4m+n).