int left[max],right[max],vis[mx];

//left[x] e rekhechi left set er x tomo node er shathe kar matching korechi

//zodi left[x]=-1 tahole ekhono karo shathe matching korate parini

vi adj[max];

bool kuhn(int u)

{

//Idea of kuhn function :

/\*

Initially karo shathe karo matching hoy ni . cnt = 0 . shob left[x] = -1 , shob right[x] = -1;

ekhon ami shob gulo left node er shathe kno 1 ta right node er matching korte chai.(tae bpm function e m ta left node er upori loop chaliyechi)

Ekhon kuhn function e ami oi node theke zeshob node e zaoa zay shegulate zacchi ebong zokhoni ekta possible matching pacchi, shei 2 ta match koriye left right update kore dicchi(ekhon ar era -1 nei).

ekhon matching 2 vabe ghotate pari ami,

1) connected kno ekta right node ekhono khali ache (-1) tahole ami easily eder matching koriye dite pari.

2) ami age kno ekta vul (non-optimal) decision nisilam zokhon left er matching koriyechi , orthat amar right node ta ekhon ze left node tar shathe matching koriyechi,hoyto oi left node take ami onno arekta right node er shathe matching korate partam ete amar matching 1 ta barto. eta korar jonno ami amar current right node visited kore dilam(porer bar ar ete zabo na karon er shathe already matching koriye felechi). tarpor ami ze left node er shathe matching koriyechi otake abar kuhn function e pathabo zodi amar current right node(occupied) chara onno karo shathe eke matching korano zeto.

zodi zay, tahole ami abar ekta matching korate parchi.

\*/

loop(x,0,SZ(adj[u])-1)

{

int v = adj[u][x];

if(vis[v]) continue;

vis[v] = 1;

if(right[v]==-1 || kuhn(right[v]))

{

right[v]=u;

left[u]=v;

return true;

}

}

return false;

}

int bpm()

{

ms(left,-1);

ms(right,-1);

int cnt = 0;

loop(x,1,m)

{

ms(vis,0);

if(kuhn(i))

cnt++:

}

return cnt;

}