```cpp
#include <bits/stdc++.h>
using namespace std;
#define ll long long

```

Check and fill bipartite graph:

```cpp
vector<int> color(100100 , -1) , adj[100100];

bool isBipartite(int src){
    queue <int> q;
    q.push(src);
    color[src] = 0;
    while(!q.empty()){
        int u = q.front();
        q.pop();
        for(auto v : adj[u]){
            if(color[v]==-1){
                color[v] = !color[u];
                q.push(v);
            }
            else if(color[v]==color[u]){
                return false;
            }
        }
    }
    return true;
}


```

Check and fill bipartite for grid:

```cpp
vector <pair<int , int>> d;
char grid[1000][1000];

void dfs(int x , int y){
    d.clear();
    if(x-1 >= 0) d.push_back({x-1 , y});
    if(x+1 < 1000) d.push_back({x+1 , y}); //check n
    if(y+1 < 1000) d.push_back({x , y+1}); //check n
    if(y-1 >= 0) d.push_back({x , y-1});
}

void isBipartite(int x , int y){
    queue <pair<int , int>> q;
    q.push({x , y});
    grid[x][y] = 'B';
    while(!q.empty()){
        auto [a , b] = q.front();
        q.pop();
        dfs(a , b);
        for(auto [i , j] : d){
            if(grid[i][j]=='.'){
                (grid[a][b]=='B' ? grid[i][j] = 'W' : grid[i][j] = 'B');
                q.push({i , j});
            }
        }
    }
}
```