int dfs(vector<vector<int> >&arr, int i, int j, int count)

{

if(i<0 || i>=arr.size() || j<0 || j>=arr[0].size() || arr[i][j] ==0)

return count;

else

{

arr[i][j] =0;

count++;

int dir[4][2] ={ {-1,0}, {1,0},{0,-1}, {0,1}};

int k;

for(k=0;k<4;k++)

{

int newi = i + dir[k][0];

int newj = j+ dir[k][1];

count = dfs(arr,newi,newj, count);

}

return count;

}

}

vector <int> countGroups(vector < vector<int> > arr, vector <int> t) {

int i , j;

map<int,int> m;

map<int,int>::iterator it;

vector<int>res;

for(i=0;i<arr.size();i++)

{

for(j=0;j<arr[0].size();j++)

{

if(arr[i][j]){

int k = dfs(arr,i,j,0);

cout<<k<<endl;

it = m.find(k);

if (it == m.end())

{

m[k] =1;

}

else

m[k]++;

}

}

}

for(i=0;i<t.size();i++)

{

if(m.find(t[i]) != m.end())

res.push\_back(m[t[i]]);

else

res.push\_back(0);

}

return res;

}