

```

#include <iostream>
#include <cstring>
#include <queue>
#include <cstdio>

using namespace std;

#define MAX 30

int visited[MAX];
string matrizona[MAX];
#define ii pair<int,int>
int n;

void bfs(int x,int y){

    queue<ii> fila;
    fila.push(ii(x,y));

    while(!fila.empty()){

        int aux = fila.front().first;
        int aux2 = fila.front().second;
        fila.pop();

        for (int i = -1 + aux; i < 2 + aux; ++i)
        {
            for (int j = -1 + aux2; j < aux2 + 2; ++j)
            {
                if(j>=0 && i>=0 && i<n && j<n){
                    if(matrizona[i][j] == '1'){
                        matrizona[i][j] = '0';
                        fila.push(make_pair(i,j));
                    }
                }
            }
        }
    }
}

int main(){

    int count,test = 1;

    while(cin >> n){

        count = 0;

        for (int i = 0; i < n; ++i)
        {
            cin >> matrizona[i];
        }

        for (int i = 0; i < n; ++i)
        {

```

```

    for (int j = 0; j < n; ++j)
    {

        if(matrizona[i][j] == '1'){
            count++;
            bfs(i,j);
        }

    }

    printf("Image number %d contains %d war eagles.\n",test,count);
    test++;

}

return 0;
}

```

1317

```

#include <iostream>
#include <cstdio>
#include <vector>
#include <queue>
#include <cstring>
#include <string>

using namespace std;

#define MAX 21
#define MAX2 101

vector <int> vetorzao[MAX];
vector <string> vetorSolucao[MAX];

int main(){

    int nPessoas,p1,p2,t1,t2;
    string palavra,w1,w2,w3,pa;
    vector<string> nomes;
    queue<int> fila;
    int vetor[MAX];

    memset(vetor,0,sizeof vetor);

    scanf("%d",&nPessoas);
    while(nPessoas != 0){

        for (int i = 1; i <= nPessoas; ++i)
        {
            scanf("%d",&p2);
            while(p2 != 0){

```

```

    // printf("%d\n",p2);
    vetorzao[i].push_back(p2);
    scanf("%d",&p2);
}

}

int quantidadeSpams;

scanf("%d",&p1);
while(p1 != 0){

    scanf("%d %d",&t1,&t2);

    cin >> w1 >> w2 >> w3;

    fila.push(p1);
    vetor[p1] = 1;

    while(!fila.empty()){

        quantidadeSpams = vetorzao[p1].size();

        if(quantidadeSpams < t1){
            vetorSolucao[p1].push_back(w1);
        }else if(quantidadeSpams >= t1 && quantidadeSpams < t2){
            vetorSolucao[p1].push_back(w2);
        }else{
            vetorSolucao[p1].push_back(w3);
        }

        for (int k = 0; k < quantidadeSpams; ++k)
        {
            if(vetor[vetorzao[p1][k]] == 0){
                vetor[vetorzao[p1][k]] = 1;
                fila.push(vetorzao[p1][k]);
            }
        }

        fila.pop();
        p1 = fila.front();

    }

    for (int o = 1; o <= nPessoas; ++o)
    {
        if(vetor[o] == 0){
            vetorSolucao[o].push_back(w1);
        }
    }

    memset(vetor,0,sizeof vetor);
    scanf("%d",&p1);
}

for (int q = 0; q < nPessoas; ++q)
{
    cin >> pa;
    nomes.push_back(pa);
}

```

```

}

for (int b = 1; b <= nPessoas; ++b)
{
    printf("%s: ",nomes[b-1].c_str());
    for (int c = 0; c < vetorSolucao[b].size(); ++c)
    {
        printf("%s ",vetorSolucao[b][c].c_str());
    }
    printf("\n");
}

//limpando dados;
nomes.clear();

for (int h = 1; h <= MAX; ++h)
{
    vetorzao[h].clear();
    vetorSolucao[h].clear();
}
memset(vetor,0,sizeof vetor);

scanf("%d",&nPessoas);
}
return 0;
}

```

1469

```

#include <iostream>
#include <cstdio>
#include <vector>
#include <queue>
#include <cstring>
#include <map>
#include <stack>

using namespace std;

#define MAX 501
int source;
int menorIdadeSol = 500;
vector <int> vetorzao[MAX];
int idades[MAX];
int vetor[MAX],chegouUmaVez = 0;
map<int,int> localizador;
stack<pair<int,int > > pilha;
int menor = 500;
int respostaFinal = 500;

int main(){

    int empregados,gerenciasDiretas,instrucoes,idade;

```

```

while(scanf("%d %d %d",&empregados,&gerenciasDiretas,&instrucoes)!=EOF){

for (int i = 1; i <= empregados; ++i)
{
    scanf("%d",&idade);
    idades[i] = idade;
}

int e1,e2;

for (int o = 1; o <= gerenciasDiretas; ++o){

    scanf("%d %d",&e1,&e2);
    vetorzao[e2].push_back(e1);

}

// for (int j = 1; j <= empregados; ++j)
// {
//     for(auto k : vetorzao[j]){

//         printf("Inicio %d fim %d\n",j,k );

//     }
// }

char letra;
memset(vetor,0,sizeof vetor);

for (int e = 0; e < instrucoes; ++e)
{

    int n1,n2,tamanhoLista;
    queue<int> fila;
    scanf(" %c",&letra);

    if(letra == 'P'){

        memset(vetor,0,sizeof vetor);
        scanf("%d",&source);

        menor = 1000;
        fila.push(source);
        vetor[source] = 1;

        while(!fila.empty()){

            int u = fila.front();fila.pop();

            tamanhoLista = vetorzao[u].size();

            for (int t = 0; t < tamanhoLista; ++t)
            {
                int v = vetorzao[u][t];
                if(vetor[v] == 0){
                    vetor[v] = 1;
                    fila.push(v);
                    // printf("Idade de v foi%d\n",idades[v]);
                    menor = min(menor,idades[v]);
                }
            }
        }
    }
}

```

```

    }

}

}

if(menor == 1000){
    printf("*\n");
}
else{
    printf("%d\n",menor);
}

}else if(letra == 'T'){

    scanf("%d %d",&n1,&n2);

    swap(vetorzao[n1],vetorzao[n2]);

    for (int i = 1; i <= empregados; ++i)
    {
        int tamanho = vetorzao[i].size();
        for(int k = 0;k < tamanho;++k){
            if(vetorzao[i][k] == n2) vetorzao[i][k] = n1;
            else if(vetorzao[i][k] == n1) vetorzao[i][k] = n2;
        }
    }

}

menor = 500;
// respostaFinal = 500;
// chegouUmaVez = 0;
memset(vetor,0,sizeof vetor);

}

for (int f = 1; f <= empregados; ++f)
{
    vetorzao[f].clear();
}

}

return 0;
}

```

1550

```

#include <iostream>
#include <cstring>
#include <queue>
#include <cstdio>

using namespace std;

```

```

#define MAX 10000

int dist[MAX];

int invertNumero(int numero){

    int reverse = 0;

    while(numero != 0) {
        int rem = numero%10;
        reverse = reverse*10 + rem;
        numero/=10;
    }

    return reverse;
}

int main(){

    int t,a,b;
    queue<int> fila;

    scanf("%d",&t);

    for (int i = 0; i < t; ++i)
    {
        memset(dist,0,sizeof dist);

        while(!fila.empty()){
            fila.pop();
        }

        scanf("%d %d",&a,&b);

        fila.push(a);

        while(!fila.empty()){

            if(dist[b] != 0){
                break;
            }

            int numero = fila.front();
            fila.pop();

            if(numero + 1 <= MAX){

                if(dist[numero + 1] == 0){

                    dist[numero + 1] = dist[numero] + 1;
                    fila.push(numero + 1);

                }

            }

            int v = invertNumero(numero);
            if(v <= MAX){

```

```

        if(dist[v] == 0){

            dist[v] = dist[numero] + 1;
            fila.push(v);

        }

    }

}

printf("%d\n",dist[b]);

}

return 0;
}

```

1835

```

#include <iostream>
#include <string>
#include <cstdio>
#include <queue>
#include <cstring>
#include <vector>

#define MAX 101
using namespace std;

vector<int> vetorzao[MAX];

int main(){
    int numerocasos,pontos,estradas,ini,destino,p1;
    int vetor[MAX];
    queue<int> fila;

    memset(vetor,0,sizeof vetor);

    scanf("%d",&numerocasos);

    for(int l = 0;l<numerocasos;++l){

        scanf("%d %d",&pontos,&estradas);

        for(int i = 0;i < estradas;++i){

            scanf("%d %d",&ini,&destino);
            vetorzao[ini].push_back(destino);
            vetorzao[destino].push_back(ini);

        }

        p1 = 1;
    }
}

```



```

fila.push(p1);
vetor[p1] = 1;
int tamanho;
int numeroBlocos = 0;

while(!fila.empty()){

    tamanho = vetorzao[p1].size();

    if(tamanho > 0){

        for(int k = 0;k<tamanho;++k){
            if(vetor[vetorzao[p1][k]] == 0){
                fila.push(vetorzao[p1][k]);
                vetor[vetorzao[p1][k]] = 1;
            }
        }

    }

    fila.pop();
    p1 = fila.front();

    if(fila.empty()){

        for (int w = 1; w <= pontos; ++w)
        {
            if(vetor[w] == 0){
                p1 = w;
                numeroBlocos++;
                vetor[w] = 1;
                fila.push(w);
                break;
            }
        }

    }

}

if(numeroBlocos > 0){
    printf("Caso #%d: ainda falta(m) %d estrada(s)\n",l + 1,numeroBlocos);
}else{
    printf("Caso #%d: a promessa foi cumprida\n",l+1);
}

memset(vetor,0,sizeof vetor);
for (int h = 1; h <= pontos; ++h)
{
    vetorzao[h].clear();
}

}

return 0;
}

```

1907

```
#include <iostream>
#include <cstdio>

using namespace std;

#define MAX 1032

int matrix[MAX][MAX];
int x,y;

void bfs(int i,int j){

    if(i > 0 && matrix[i - 1][j] == 0){
        matrix[i - 1][j] = 1;
        bfs(i-1,j);
    }

    if(i < x-1 && matrix[i + 1][j] == 0){
        matrix[i + 1][j] = 1;
        bfs(i + 1,j);
    }

    if(j > 0 && matrix[i][j - 1] == 0){
        matrix[i][j - 1] = 1;
        bfs(i,j - 1);
    }

    if(j < y - 1 && matrix[i][j + 1] == 0){
        matrix[i][j + 1] = 1;
        bfs(i,j + 1);
    }

}

int main(){

    char letra;
    scanf("%d %d",&x,&y);
    int cont = 0;

    for (int i = 0; i < x; ++i)
    {
        for (int j = 0; j < y; ++j)
        {
            scanf(" %c",&letra);
            if(letra == 'o') matrix[i][j] = 1;
        }
    }

    for (int i = 0; i < x; ++i)
    {
        for (int j = 0; j < y; ++j)
```

```

{
    if(matrix[i][j] == 0){
        cont++;
        matrix[i][j] = 1;
        bfs(i,j);
    }

}

}

printf("%d\n",cont);

return 0;
}

```

1910

```

#include <iostream>
#include <cstring>
#include <queue>
#include <cstdio>

using namespace std;
#define limite 100000
#define MAX 1000000

int dist[MAX];
int naopode[MAX];

int main(){

    int origem,destino,proibidos,canal;
    queue<int> fila;
    int flag = 0;

    scanf("%d %d %d",&origem,&destino,&proibidos);
    while(origem > 0 && destino > 0){

        memset(dist,-1,sizeof dist);
        memset(naopode,0,sizeof naopode);
        flag = 0;

        int numero;

        for (int i = 0; i < proibidos; ++i)
        {
            scanf("%d",&numero);
            naopode[numero] = 1;
        }

        fila.push(origem);
        dist[origem] = 0;

        while(!fila.empty()){

```

```

if(dist[destino] != -1){
    flag = 1;
    break;
}

canal = fila.front();
fila.pop();

if(canal + 1 <= limite){
    if(naopode[canal + 1] != 1 && dist[canal + 1] == -1){
        dist[canal + 1] = dist[canal] + 1;
        fila.push(canal + 1);
    }
}

if(canal - 1 > 0){
    if(naopode[canal - 1] != 1 && dist[canal - 1] == -1){
        dist[canal - 1] = dist[canal] + 1;
        fila.push(canal - 1);
    }
}

if(canal * 2 <= limite){
    if(naopode[canal * 2] != 1 && dist[canal * 2] == -1){
        dist[canal * 2] = dist[canal] + 1;
        fila.push(canal * 2);
    }
}

if(canal * 3 <= limite){
    if(naopode[canal * 3] != 1 && dist[canal * 3] == -1){
        dist[canal * 3] = dist[canal] + 1;
        fila.push(canal * 3);
    }
}

if(canal%2 == 0){
    if(naopode[canal/2] != 1 && dist[canal/2] == -1){
        dist[canal/2] = dist[canal] + 1;
        fila.push(canal/2);
    }
}

}

if(flag == 1){
    printf("%d\n",dist[destino]);
}else{
    printf("-1\n");
}

while(!fila.empty()){

    fila.pop();

}

scanf("%d %d %d",&origem,&destino,&proibidos);
}

```

```

    return 0;
}

// # If(dist[destino] != -1) break;

// if(canal + 1 <= limite){
//     if(naopode[canal + 1] != 1 && dist[canal + 1] == -1){
//         dist[canal + 1] = dist[canal] + 1;
//         fila.push(canal + 1);
//     }

// }

// # 12582
// # 12376

```

### 10305 com algoriimo de Khan

```

#include <iostream>
#include <vector>
#include <cstring>
#include <queue>
#include <cstdio>

using namespace std;

#define MAX 101
vector<int> vetorzao[MAX];
int visitados[MAX];
vector<int> resposta;
int grau[MAX];

int main(){

    int m,n;

    scanf("%d %d",&n,&m);

    while(m != 0 && n != 0){

        int p,q;
        memset(visitados,0,sizeof visitados);
        memset(grau,0,sizeof grau);

        for (int l = 0; l < m; ++l)
        {
            scanf("%d %d",&p,&q);
            vetorzao[p].push_back(q);
            grau[q]++;
        }
    }
}

```

```

}

int flag = 0, atual;

while(flag == 0){

    flag = 1;
    for (int i = 1; i <= n; ++i)
    {
        if(grau[i] == 0 && visitados[i] == 0){
            flag = 0;
            atual = i;
            resposta.push_back(i);
            visitados[i] = 1;
            break;
        }

    }

    for(auto des : vetorzao[atual]){

        grau[des]--;

    }

}

for (int i = 0; i < resposta.size(); ++i)
{
    if((i!=resposta.size() - 1)){
        printf("%d ",resposta[i]);
    }else{
        printf("%d\n",resposta[i]);
    }
}

printf("\n");

resposta.clear();

for (int i = 1; i <= n; ++i)
{
    vetorzao[i].clear();
}

scanf("%d %d",&n,&m);
}

return 0;
}

```

```

#include <iostream>
#include <vector>
#include <cstring>
#include <queue>
#include <cstdio>

using namespace std;

#define MAX 101
vector<int> vetorzao[MAX];
int visitados[MAX];
vector<int> resposta;

void dfs(int n){

    if(visitados[n] == 1){
        return;
    }

    visitados[n] = 1;

    for(auto v : vetorzao[n]){

        if(visitados[v] == 0){
            dfs(v);
        }

    }

    resposta.push_back(n);
}

int main(){

    int m,n;

    scanf("%d %d",&n,&m);

    while(m != 0 && n != 0){

        int p,q;
        memset(visitados,0,sizeof visitados);

        for (int l = 0; l < m; ++l)
        {
            scanf("%d %d",&p,&q);
            vetorzao[p].push_back(q);
        }

        for (int i = 1; i <= n; ++i)
        {
            if(visitados[i] == 0){
                dfs(i);
            }
        }

        for (int i = (resposta.size() - 1); i >= 0; --i)
        {

```

```

    if(i!=0){
        printf("%d ",resposta[i]);
    }else{
        printf("%d",resposta[i]);
    }
}

printf("\n");

resposta.clear();
for (int i = 1; i <= n; ++i)
{
    vetorao[i].clear();
}
scanf("%d %d",&n,&m);
}

return 0;
}

```

11244

```

#include <iostream>
#include <cstdio>
#include <vector>
#include <string>
#include <cstring>
#include <queue>

using namespace std;

#define MAX 101
#define ii pair<int,int>
string matrizona[MAX];
int cont = 0,linha,coluna;
int visited[MAX];

void dfs(int x,int y){

    queue<ii> fila;
    fila.push(ii(x,y));

    while(!fila.empty()){

        int aux = fila.front().first;
        int aux2 = fila.front().second;
        fila.pop();

        for (int i = -1 + aux; i < 2 + aux; ++i)
        {
            for (int j = -1 + aux2; j < aux2 + 2; ++j)
            {
                if(j>=0 && i>=0 && i<linha && j<coluna){

```



```

        if(matrizona[i][j] == '*'){
            matrizona[i][j] = '.';
            cont++;
            fila.push(make_pair(i,j));
        }
    }
}
}

}

}

int main(){

    int estrelas = 0;

    while(scanf("%d %d",&linha,&coluna),linha | coluna){

        memset(visited,0,sizeof visited);
        memset(visited,0,sizeof visited);

        for (int i = 0; i < linha; ++i)
        {
            cin >> matrizona[i];
        }

        for (int i = 0; i < linha; ++i)
        {

            for (int j = 0; j < coluna; ++j)
            {

                if(matrizona[i][j] == '*'){
                    dfs(i,j);
                    if(cont == 1){
                        estrelas++;
                    }
                    cont = 0;
                }

            }
        }

        printf("%d\n",estrelas);
        estrelas = 0;

    }

    return 0;
}

```

## BFS

```
#include <iostream>
```

```

#include <cstdio>
#include <vector>
#include <queue>
#include <cstring>

using namespace std;

#define MAX 101

vector<int> vetorzao[MAX];
int vetor[MAX];

int main(){

    int inicio,destino,noInicial,n2;
    queue<int> fila;
    vector<int> naoPassou;

        memset(vetor,0,sizeof vetor);

    scanf("%d",&n2);
    while(n2 != 0){

        scanf("%d",&inicio);
        while(inicio != 0){

            scanf("%d",&destino);
            while(destino != 0){

                vetorzao[inicio].push_back(destino);
                scanf("%d",&destino);

            }

            scanf("%d",&inicio);
        }

        int tamanho,qtsNosIniciais,noDestino;

        scanf("%d",&qtsNosIniciais);

        for (int i = 0; i < qtsNosIniciais; ++i)
        {

            scanf("%d",&noInicial);

            fila.push(noInicial);

            while(!fila.empty()){

                tamanho = vetorzao[noInicial].size();

                for (int k = 0; k < tamanho; ++k)
                {
                    if(vetor[vetorzao[noInicial][k]] == 0){
                        vetor[vetorzao[noInicial][k]] = 1;
                        fila.push(vetorzao[noInicial][k]);
                    }
                }
            }
        }
    }
}

```

```

        fila.pop();
        noInicial = fila.front();

    }

    int contador = 0;

    for (int l = 1; l <= n2; ++l)
    {
        if(vetor[l] == 0){
            contador++;
            naoPassou.push_back(l);
        }
    }

    printf("%d",contador);

    for (int j = 0; j < naoPassou.size(); ++j)
    {
        printf(" %d",naoPassou[j]);
    }
    printf("\n");

    naoPassou.clear();
    memset(vetor,0,sizeof vetor);

}

for (int z = 1; z <= n2; ++z)
{
    vetorzao[z].clear();
}

scanf("%d",&n2);

}

return 0;
}

```

## DFS

```

#include <iostream>
#include <cstdio>
#include <vector>
#include <cstring>
#include <memory>

using namespace std;

#define MAX 50010

int vetorzao[MAX];
int vetor[MAX],dist[MAX];

```

```

int contador = 0,pontoDePartida;

void dfs(int source,int size){
    if(vetor[source]){
        return;
    }
    vetor[source] = 1;
    int v = vetorzao[source];

    if(dist[v]){
        dist[source] = dist[v] + 1;
    }
    if(vetor[v] == 0){

        dfs(v,size+1);
        if(!dist[source]){
            dist[source] = dist[v] + 1;
        }

    }else{

        while(dist[source] == 0){

            dist[source] = size;
            source = vetorzao[source];

        }

    }

}

int main(){

    int n,inicio,destino,casosDeTeste;

    scanf("%d",&casosDeTeste);

    for (int f = 0; f < casosDeTeste; ++f)
    {
        scanf("%d",&n);

        memset(vetor,0,sizeof vetor);
        memset(dist,0,sizeof dist);
        memset(vetorzao,0,sizeof vetorzao);

        for (int i = 0; i < n; ++i)
        {
            scanf("%d %d",&inicio,&destino);
            vetorzao[inicio] = destino;
        }

        for(int j = 1;j <= n;++j){

            if(vetor[j] == 0) dfs(j,1);

        }

        int maior = -1,posicao;
    }

```

```

    for (int k = 1; k <= n; ++k)
    {
        if(dist[k] > maior){
            maior = dist[k];
            posicao = k;
        }
    }

    printf("Case %d: %d\n",f + 1,posicao);

}

return 0;
}

```

11060

```

#include <iostream>
#include <vector>
#include <cstring>
#include <queue>
#include <map>
#include <string>

using namespace std;

#define MAX 110

//procurar os verticos com chegada igual a zero. quando se repete, pegar na ordem da entrada.
//usar fila de prioridades ao invés de uma fila comum
// int adj[MAX][MAX];
// int in[MAX],out[MAX];
vector<int> vetorzao[MAX];
int visitados[MAX];
int grau[MAX];
map<string,int> mapa;
map<int,string> volta;
vector<string> resposta;

int main(){

    int n,caso = 1;

    while(scanf("%d",&n) == 1){

        string palavra,inicio,destino;
        memset(visitados,0,sizeof visitados);
        memset(grau,0,sizeof grau);

        for (int i = 1; i <= n; ++i)
        {
            cin >> palavra;
            mapa[palavra] = i;
            volta[i] = palavra;
        }
    }
}

```

```

int m,a,b;
scanf("%d",&m);

for(int k = 0;k < m;++k){

    cin >> inicio >> destino;

    a = mapa[inicio];
    b = mapa[destino];
    vetorzao[a].push_back(b);
    grau[b]++;

}

int flag = 0,atual;

while(flag == 0){

    flag = 1;
    for (int i = 1; i <= n; ++i)
    {
        if(grau[i] == 0 && visitados[i] == 0){
            flag = 0;
            atual = i;
            resposta.push_back(volta[i]);
            visitados[i] = 1;
            break;
        }

    }

    for(auto des : vetorzao[atual]){

        grau[des]--;

    }

}

printf("Case #%d: Dilbert should drink beverages in this order:",caso);

for (int y = 0; y <resposta.size(); ++y)
{
    printf(" %s",resposta[y].c_str());
}

printf("\n");
printf("\n");

resposta.clear();
mapa.clear();
volta.clear();
caso++;

for (int i = 1; i <= n; ++i)
{
    vetorzao[i].clear();
}

```

```
}
```

```
return 0;
```

```
}
```