Q1

```
#include <bits/stdc++.h>
using namespace std;
const int INF = 1e9;
typedef pair<int, int> pii;
typedef vector<pii> vii;
typedef vector<int> vi;

void init(vector<int>& dis, vector<int>& vis) {
    fill(dis.begin(), dis.end(), -1);
    fill(vis.begin(), vis.end(), -1);
}
int main() {
    int n, m;
    cin >> n >> m;
    vector<vii> graph(n + 1);
    for (int i = 0; i < m; i++) {
        int a, b, c;
        cin >> a >> b >> c;
        graph[a].push_back({ c, b });
    }
    priority_queue<pii, vii, greater<pii>> q;
    q.push({ 0, 1 });

    vi dis(n + 1);
    vi vis(n + 1);

    init(dis, vis);
    dis[1] = 0;
    while (!q.empty()) {
        pii p = q.top();
        q.pop();
        int node = p.second;
        int d = p.first;

        if (vis[node] != -1) continue;
        vis[node] = 1;

        for (pii aa : graph[node]) {
            if (vis[aa.second] != -1) continue;
            if (dis[aa.second] == -1 || dis[aa.second] > d + aa.first) {
                dis[aa.second] = d + aa.first;
                q.push({ d + aa.first, aa.second });
            }
        }
    }

    for (int i = 1; i <= n; i++) {
        cout << dis[i] << " ";
    }
    cout << endl;

    return 0;
}
```

Q2

```
#include <bits/stdc++.h>
```

```
using namespace std;

const int inf = 1e9;

#define fo(i,a,b) for(int i=a;i<b;i++)

int main() {
    int n, m, q;
    cin >> n >> m >> q;

    vector<vector<int>> mat(n + 1, vector<int>(n + 1, inf));

    fo(i, 1, n + 1) {
        fo(j, 1, n + 1) {
            mat[i][j] = inf;
        }
        mat[i][i] = 0;
    }

    fo(i, 0, m) {
        int a, b, c;
        cin >> a >> b >> c;
        mat[a][b] = min(mat[a][b], c);
        mat[b][a] = min(mat[b][a], c);
    }

    fo(k, 1, n + 1) {
        fo(i, 1, n + 1) {
            fo(j, 1, n + 1) {
                mat[i][j] = min(mat[i][j], mat[i][k] + mat[k][j]);
            }
        }
    }

    fo(i, 0, q) {
        int a, b;
        cin >> a >> b;
        cout << ((mat[a][b] >= inf) ? -1 : mat[a][b]) << "\n";
    }

    return 0;
}
```

Q3

```
#include <bits/stdc++.h>
using namespace std;
bool is_valid(long long i, long long j, long long n)
{
    if (i < 0 || j < 0 || i >= n || j >= n)
        return false;
    return true;
}
int main()
{
    long long n;
    cin >> n;
    vector<vector<long long>> a(n, vector<long long>(n, 0));
    for (long long i = 0; i < n; i++)
        for (long long j = 0; j < n; j++)
            cin >> a[i][j];
    vector<pair<long long, long long>> adj[n * n + 1];
    for (long long i = 0; i < n; i++)
    {
        for (long long j = 0; j < n; j++)
        {
            if (is_valid(i, j - 1, n))
                adj[i * n + j + 1].push_back({a[i][j - 1], i * n + j});
            if (is_valid(i, j + 1, n))
                adj[i * n + j + 1].push_back({a[i][j + 1], i * n + j + 2});
            if (is_valid(i - 1, j, n))
                adj[i * n + j + 1].push_back({a[i - 1][j], (i - 1) * n + j + 1});
            if (is_valid(i + 1, j, n))
                adj[i * n + j + 1].push_back({a[i + 1][j], (i + 1) * n + j + 1});
        }
    }
    priority_queue<pair<long long, long long>, vector<pair<long long, long long>>,
greater<pair<long long, long long>>> pq;
    pq.push({a[0][0], 1});
    vector<bool> vis(n * n + 1, false);
    vector<long long> dis(n * n + 1, LONG_MAX);
    dis[1] = a[0][0];
    while (!pq.empty())
    {
        pair<long long, long long> x = pq.top();
        dis[x.second] = x.first;
        vis[x.second] = true;
        pq.pop();
        for (auto j : adj[x.second])
        {
            if (!vis[j.second] && dis[j.second] > x.first + j.first)
            {
                dis[j.second] = dis[x.second] + j.first;
                pq.push({dis[j.second], j.second});
            }
        }
    }
    cout << dis[n * n] << endl;
    return 0;
}
```