```
// Dijkstra con potenciales - 0.04s
int cost[50][50], cap[50][50], flow[50][50], s, t, V, inf = 1 << 29, par[50], dist[50], p[50];

inline int ecap(int a, int b){
    if(flow[b][a]) return flow[b][a];
    else return cap[a][b] - flow[a][b];
}

inline int ecost(int a, int b){
    if(flow[b][a]) return -cost[b][a] + p[a] - p[b];
    else return cost[a][b] + p[a] - p[b];
}

bool seen[50];
bool augment(){
    for(int i = 0; i < V; i++) par[i] = -1, dist[i] = inf;
    dist[s] = 0; par[s] = -2;

    memset(seen, 0, sizeof seen);
    int u = s;
    while(u != -1){
        seen[u] = true;
        for(int v = 0; v < V; v++)
        if(ecap(u, v) && dist[v] > ecost(u, v) + dist[u])
            dist[v] = dist[u] + ecost(u, v), par[v] = u;
        u = -1;
        for(int i = 0; i < V; i++) if(!seen[i] && dist[i] != inf && (u == -1 || dist[u] > dist[i])) u = i;
    }
    for(int v = 0; v < V; v++) if(dist[v] != inf) p[v] += dist[v];
    return dist[t] != inf;
}

int mcmf(){
    int res = 0;
    memset(p, 0, sizeof p);
    while(augment()){
        for(int v = t, u = par[v]; u != -2; u = par[v = u])
            if(flow[v][u]) flow[v][u]--, res -= cost[v][u];
            else flow[u][v]++, res += cost[u][v];
    }
    return res;
}

int main(){
    int y[16][2];
    int m;
    while(scanf("%d", &m) && m){
        for(int i = 0; i < m; i++) scanf("%d", &y[i][0]);
        for(int i = 0; i < m; i++) scanf("%d", &y[i][1]);
        s = m * 2; t = s + 1; V = t + 1;
        memset(cap, 0, sizeof cap);
        memset(cost, 0, sizeof cost);
        memset(flow, 0, sizeof flow);
        for(int i = 0; i < m; i++) for(int j = 0; j < m; j++)
            cap[i][j + m] = 1, cost[i][j + m] = abs(i - j) + abs(y[i][0] - y[j][1]);
        for(int i = 0; i < m; i++) cap[s][i] = cap[i + m][t] = 1;
        cout << mcmf() << endl;
    }
}
```

```
// Dijkstra sin potenciales - 0.03s
int cost[50][50], cap[50][50], flow[50][50], s, t, V, inf = 1 << 29, par[50], dist[50];

inline int ecap(int a, int b){
    if(flow[b][a]) return flow[b][a];
    else return cap[a][b] - flow[a][b];
}

inline int ecost(int a, int b){
    if(flow[b][a]) return -cost[b][a];
    else return cost[a][b];
}

bool seen[50];
bool augment(){
    for(int i = 0; i < V; i++) par[i] = -1, dist[i] = inf;
    dist[s] = 0; par[s] = -2;
    memset(seen, 0, sizeof seen);
    int u = s;
    while(u != -1){
        seen[u] = true;
        for(int v = 0; v < V; v++)
        if(ecap(u, v) && dist[v] > dist[u] + ecost(u, v)){
            dist[v] = dist[u] + ecost(u, v);
            par[v] = u;
            seen[v] = false;
        }
        u = -1;
        for(int i = 0; i < V; i++) if(!seen[i] && dist[i] != inf && (u == -1 || dist[u] > dist[i])) u = i;
    }
    return dist[t] != inf;
}

int mcmf(){
    int res = 0;
    while(augment()){
        for(int v = t, u = par[v]; u != -2; u = par[v = u])
            if(flow[v][u]) flow[v][u]--, res -= cost[v][u];
            else flow[u][v]++, res += cost[u][v];
    }
    return res;
}

int main(){
    int y[16][2];
    int m;
    while(scanf("%d", &m) && m){
        for(int i = 0; i < m; i++) scanf("%d", &y[i][0]);
        for(int i = 0; i < m; i++) scanf("%d", &y[i][1]);
        s = m * 2; t = s + 1; V = t + 1;
        memset(cap, 0, sizeof cap);
        memset(cost, 0, sizeof cost);
        memset(flow, 0, sizeof flow);
        for(int i = 0; i < m; i++) for(int j = 0; j < m; j++)
            cap[i][j + m] = 1, cost[i][j + m] = abs(i - j) + abs(y[i][0] - y[j][1]);
        for(int i = 0; i < m; i++) cap[s][i] = cap[i + m][t] = 1;
        cout << mcmf() << endl;
    }
}
```

```
// Bellman-Ford - 0.04s
int cost[50][50], cap[50][50], flow[50][50], s, t, V, inf = 1 << 29, par[50], dist[50];

inline int ecap(int a, int b){
    if(flow[b][a]) return flow[b][a];
    else return cap[a][b] - flow[a][b];
}

inline int ecost(int a, int b){
    if(flow[b][a]) return -cost[b][a];
    else return cost[a][b];
}

bool augment(){
    for(int i = 0; i < V; i++) par[i] = -1, dist[i] = inf;
    dist[s] = 0; par[s] = -2;
    bool changed = true;
    while(changed){
        changed = false;
        for(int u = 0; u < V; u++) if(dist[u] != inf) for(int v = 0; v < V; v++)
            if(ecap(u, v) && dist[v] > dist[u] + ecost(u, v)){
                dist[v] = dist[u] + ecost(u, v);
                par[v] = u;
                changed = true;
            }
    }
    return dist[t] != inf;
}

int mcmf(){
    int res = 0;
    while(augment()){
        for(int v = t, u = par[v]; u != -2; u = par[v = u])
            if(flow[v][u]) flow[v][u]--, res -= cost[v][u];
            else flow[u][v]++, res += cost[u][v];
    }
    return res;
}

int main(){
    int y[16][2];
    int m;
    while(scanf("%d", &m) && m){
        for(int i = 0; i < m; i++) scanf("%d", &y[i][0]);
        for(int i = 0; i < m; i++) scanf("%d", &y[i][1]);
        s = m * 2; t = s + 1; V = t + 1;
        memset(cap, 0, sizeof cap);
        memset(cost, 0, sizeof cost);
        memset(flow, 0, sizeof flow);
        for(int i = 0; i < m; i++) for(int j = 0; j < m; j++)
            cap[i][j + m] = 1, cost[i][j + m] = abs(i - j) + abs(y[i][0] - y[j][1]);
        for(int i = 0; i < m; i++) cap[s][i] = cap[i + m][t] = 1;
        cout << mcmf() << endl;
    }
}
```