Program:

```
import java.io.*;
import java.lang.*;
import java.util.*;
import java.util.LinkedList;
public class MaxFlow {
    static final int V = 6;
    boolean bfs(int rGraph[][], int s, int t, int parent[])
    {
        boolean visited[] = new boolean[V];
        for (int i = 0; i < V; ++i)
            visited[i] = false;
        LinkedList<Integer> queue
            = new LinkedList<Integer>();
        queue.add(s);
        visited[s] = true;
        parent[s] = -1;
        while (queue.size() != 0) {
            int u = queue.poll();
            for (int v = 0; v < V; v++) {
                if (visited[v] == false
                    && rGraph[u][v] > 0) {
                    if (v == t) {
                        parent[v] = u;
                        return true;
                    }
                    queue.add(v);
                    parent[v] = u;
                    visited[v] = true;
                }
            }
        }
        return false;
    }
    int fordFulkerson(int graph[][], int s, int t)
    {
        int u, v;
        int rGraph[][] = new int[V][V];
```

```
        for (u = 0; u < V; u++)
            for (v = 0; v < V; v++)
                rGraph[u][v] = graph[u][v];
        int parent[] = new int[V];
        int max_flow = 0;
        while (bfs(rGraph, s, t, parent)) {
            int path_flow = Integer.MAX_VALUE;
            for (v = t; v != s; v = parent[v]) {
                u = parent[v];
                path_flow
                    = Math.min(path_flow, rGraph[u][v]);
            }
            for (v = t; v != s; v = parent[v]) {
                u = parent[v];
                rGraph[u][v] -= path_flow;
                rGraph[v][u] += path_flow;
            }
            max_flow += path_flow;
        }
        return max_flow;
    }
    public static void main(String[] args)
        throws java.lang.Exception
    {
        int graph[][] = new int[][] {
            { 0, 16, 13, 0, 0, 0 }, { 0, 0, 10, 12, 0, 0 },
            { 0, 4, 0, 0, 14, 0 }, { 0, 0, 9, 0, 0, 20 },
            { 0, 0, 0, 7, 0, 4 }, { 0, 0, 0, 0, 0, 0 }
        };
        MaxFlow m = new MaxFlow();
        System.out.println("The maximum possible flow is "+ m.fordFulkerson
            (graph, 0, 5));
    }
}
```

Output: