//dijkstra

public class Main {

    public static int dijkstra(int[][] graph, int source, int destination){

        int minDistance;

        int vertices = graph.length;

        int[] distances = new int[vertices];

        boolean[] visited = new boolean[vertices];

        for(int i=0;i<vertices;i++){

            distances[i] = Integer.MAX\_VALUE;

        }

        distances[source]=0;

        for(int i=0;i<vertices;i++){

            int minVertex = findMinDistance(distances,visited);

            visited[minVertex] = true;

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

                if(!visited[j] && graph[minVertex][j] !=0 && distances[minVertex]!=Integer.MAX\_VALUE

                && distances[minVertex] + graph[minVertex][j]<distances[j]) {

                    distances[j] = distances[minVertex] + graph[minVertex][j];

                }

            }

        }

        minDistance = distances[destination];

        return minDistance;

    }

    static int findMinDistance(int[] distances,boolean visited[]){

        int minDistance = Integer.MAX\_VALUE;

        int minVertex = -1;

        for(int i=0;i<distances.length;i++){

            if(!visited[i] && distances[i]<minDistance){

                minDistance = distances[i];

                minVertex = i;

            }

        }

        return minVertex;

    }

    public static void main(String args[]){

        int[][] graph = {

            {0,1,3,0},

            {1,0,2,4},

            {3,2,0,6},

            {0,4,6,0},

        };

        int source=0;

        int destination = 3;

        int shortestDistance = dijkstra(graph,source,destination);

        System.out.println(shortestDistance);

    }

}

//bellman ford

public class Main {

    static int bellmanFord(int[][] graph, int source, int destination){

        int vertices = graph.length;

        int[] distances = new int[vertices];

        for(int i=0;i<vertices;i++){

            distances[i] = Integer.MAX\_VALUE;

        }

        distances[source] = 0;

        //bellman ford code

        for(int i=0;i<vertices-1;i++){

            for(int u=0;u<vertices;u++){

                for(int v=0;v<vertices;v++){

                    if(graph[u][v]!=0 && distances[u]!=Integer.MAX\_VALUE && distances[u]+graph[u][v] < distances[v]){

                        distances[v] = distances[u] + graph[u][v];

                    }

                }

            }

        }

        return distances[destination];

    }

    public static void main(String args[]){

        int[][] graph = {

            {0,-2,0,1},

            {0,0,4,0},

            {0,0,0,0},

            {0,-1,2,0}

        };

        int source=0;

        int destination = 2;

        int shortestDistance = bellmanFord(graph,source,destination);

        System.out.println(shortestDistance);

    }

}