

## Computer Networks Lab

**Program 2:** Write a program to find the shortest path vertices using Bellman-Ford Algorithm.

**Code:**

```
package src;
import java.util.Scanner;
public class p8 {
    private int d[], num_var;
    private static final int max_value = 999;
    public p8(int num_var){
        this.num_var = num_var;
        d = new int[num_var+1];
    }
    public void bellmanFordEvaluation(int source, int a[][]) {
        for(int node=1;node<=num_var;node++)
            d[node] = max_value;
        d[source]=0;
        for(int node=1;node<=num_var-1;node++)
            for(int sn=1;sn<=num_var;sn++)
                for(int dn=1;dn<=num_var;dn++)
                    if(a[sn][dn]!=max_value)
                        if(d[dn]>d[sn]+a[sn][dn])
                            d[dn] = d[sn] + a[sn][dn];
        for(int sn=1;sn<=num_var;sn++)
            for(int dn=1;dn<num_var;dn++)
                if(a[sn][dn]!=max_value)
                    if(d[dn]>d[sn]+a[sn][dn])
                        System.out.println("The graph contains
negative edge cycle");
        for(int vertex=1;vertex<=num_var;vertex++)
            System.out.println("Distance of source "+source+" to
"+vertex+" is "+d[vertex]);
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
```

```
int num_var=0, source;
Scanner sc = new Scanner(System.in);
System.out.println("Enter the number of vertices: ");
num_var = sc.nextInt();
int a[][] = new int[num_var+1][num_var+1];
System.out.println("Enter the adjacency matrix:");
for(int sn=1;sn<=num_var;sn++)
    for(int dn=1;dn<=num_var;dn++) {
        a[sn][dn] = sc.nextInt();
        if(sn==dn) {
            a[sn][dn]=1;
            continue;
        }
        if(a[sn][dn]==0)
            a[sn][dn] = max_value;
    }
System.out.println("Enter the source vertex:");
source = sc.nextInt();
p8 b = new p8(num_var);
b.bellmanFordEvaluation(source, a);
sc.close();
}
}
```

### Output:

Enter the number of vertices:

4

Enter the adjacency matrix:

0 5 0 0

5 0 3 4

0 3 0 2

0 4 2 0

Enter the source vertex:

1

Distance of source 1 to 1 is 0

Distance of source 1 to 2 is 5

Dept of ISE, RNSIT

Distance of source 1 to 3 is 8

Distance of source 1 to 4 is 9