Name – Saikat Sheet

University Roll – 18700120024.

Department – CSE(A)

Assignment-8

Code:

#include <stdio.h>

#include <stdlib.h>

#define INT\_MAX 999

#define NIL -1

int d[5];

int p[5];

int n, s;

int queue;

int w[5][5];

int done[5];

void relax(int u, int v)

{

if(d[v] > d[u] +w[u][v])

{

d[v] = d[u] +w[u][v];

p[v] = u;

}

}

void initalise\_source()

{

queue = n;

for(int i = 0; i<n; i++)

{

d[i] = INT-MAX;

p[i] = NIL;

done[i] = 0;

}

d[s] = 0;

}

int extract\_min()

{

int minimum = INT\_MAX;

int m = -1;

for(int i = 0; i<n; i++)

{

if(!done[i])

{

if(d[i] <= minimum)

{

minimum = d[i];

m = i;

}

}

}

done[m] = 1;

return m;

}

void dijkstra()

{

initalise\_source();

while(queue != 0)

{

int u = extract\_min();

for(int i = 0; i<n; i++)

{

if(w[u][i] != INT\_MAX)

relax(u, i);

}

queue--;

}

}

int main()

{

printf("Enter the size of array: ");

scanf("%d", &n);

printf("Enter the matrix of graph\n");

for(int i = 0; i<n; i++)

{

for(int j = 0; j<n; j++)

{

scanf("%d", &w[i][j]);

}

}

printf("Enter initialize source: ");

scanf("%d", &s);

dijkstra();

for(int i = 0; i<n; i++)

printf("%d ", d[i]);

}

Output:

