

PROGRAM:

```
#include <stdio.h>

#include <stdbool.h>

int D[100], P[100];
int n, adj[100][100];
int edges[100][3];
int source;
int e = 0;

void initialize(), printEstimates(), printPredecessors();
void relax(int, int, int);
void printPath(int);

bool noNegCycles();

int main() {
    printf("Enter the number of vertices : ");
    scanf("%d", &n);

    printf("Enter the adjacency matrix : \n");
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            scanf("%d", &adj[i][j]);
        }
    }

    printf("The graph is : \n");
    for(int i = 0; i < n; i++) {
```

```

    for(int j = 0; j < n; j++) {
        printf("%d ", adj[i][j]);
    }
    printf("\n");
}

```

```

printf("\nThe edges are : \n");
for(int i = 0; i < n; i++) {
    for(int j = 0; j < n; j++) {
        if(adj[i][j] > 0) {
            printf("%d -> %d : %d\n", i, j, adj[i][j]);
            edges[e][0] = i;
            edges[e][1] = j;
            edges[e][2] = adj[i][j];
            e++;
        }
    }
}

```

```

printf("\nEnter source vertex : ");
scanf("%d", &source);

```

```

printf("Initializing...\n");
initialize();
printEstimates();
printPredecessors();

```

```

printf("\nRunning Bellman Ford Algorithm...\n");
for(int i = 1; i <= n-1; i++) {

```

```

    for(int j = 0; j < e; j++) {
        relax(edges[j][0], edges[j][1], edges[j][2]);
    }
}

printEstimates();
printPredecessors();

if(!noNegCycles()) {
    printf("\nSSSP does NOT exist! Negative cycle found!");
}
else {
    printf("\nSSSP Exists. The paths are : \n");
    for(int i = 0; i < n; i++) {
        if(i != source) {
            printPath(i);
            printf("\n");
        }
    }
}

return 0;
}

void relax(int u, int v, int w) {
    if(D[u] + w < D[v]) {
        D[v] = D[u] + w;
        P[v] = u;
    }
}

```

```
    }  
}
```

```
void initialize() {  
    for(int i = 0; i < n; i++) {  
        D[i] = 999;  
        P[i] = -1;  
    }  
    D[source] = 0;  
}
```

```
void printEstimates() {  
    printf("\nThe shortest path estimates are : \n");  
    for(int i = 0; i < n; i++) {  
        printf("%d ", D[i]);  
    }  
}
```

```
void printPredecessors() {  
    printf("\nThe predecessors are : \n");  
    for(int i = 0; i < n; i++) {  
        printf("%d ", P[i]);  
    }  
}
```

```
bool noNegCycles() {  
    for(int i = 0; i < e; i++) {  
        int u = edges[i][0], v = edges[i][1], w = edges[i][2];  
        if(D[u] + w < D[v]) {
```

```
        return false;
    }
}

return true;
}

void printPath(int v) {
    if(v == source) {
        printf("%d -> ", source);
    }
    else if(P[v] == -1) {
        printf("NO PATH");
    }
    else {
        printPath(P[v]);
        printf("%d -> ", v);
    }
}
```

OUTPUT:

```
PS D:\Harsh\SEM 4\AOA\Assignment\Assign 6> cd "d:\Harsh\SEM 4\AOA\Assignment\Assign 6\" ; if ($?) { .
Enter the number of vertices : 4
Enter the adjacency matrix :
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
The graph is :
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

The edges are :
0 -> 0 : 1
0 -> 1 : 2
0 -> 2 : 3
0 -> 3 : 4
1 -> 0 : 5
1 -> 1 : 6
1 -> 2 : 7
1 -> 3 : 8
2 -> 0 : 9
2 -> 1 : 10
2 -> 2 : 11
2 -> 3 : 12
3 -> 0 : 13
3 -> 1 : 14
3 -> 2 : 15
3 -> 3 : 16

Enter source vertex : 0
Initializing...

The shortest path estimates are :
0 999 999 999
The predecessors are :
-1 -1 -1 -1
Running Bellman Ford Algorithm...

The shortest path estimates are :
0 2 3 4
The predecessors are :
-1 0 0 0
SSSP Exists. The paths are :
0 -> 1 ->
0 -> 2 ->
0 -> 3 ->
PS D:\Harsh\SEM 4\AOA\Assignment\Assign 6> █
```