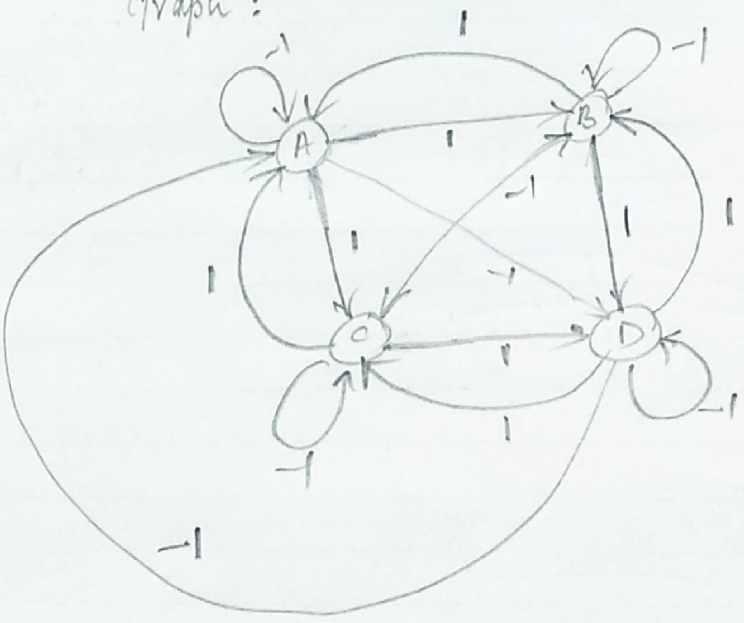
Output:

```
$ ./dv
Enter the no. of vertices : 4
Enter the adjacency matrix
-1  1  1  -1
 1 -1 -1   1
 1 -1 -1   1
-1  1  1  -1
Negative edge
```

Execution example for a graph with negative edge weights

Graph :



Another execution instance in the next page.

2. Write a program to implement distance vector routing protocol for a simple topology of routers.

```
#include <stdio.h>
#include <stdlib.h>

int A[10][10], n, d[10], p[10];

void BellmanFord( int s)
{
        int i, u, v;
        for( i=1 ; i<n; i++) {
            for( u=0 ; u<n ; u++)
                for ( v=0 ; v<n ; v++)
                    if ( d[v] >d[u] + A[u][v] ) {
                        d[v] = d[u] + A[u][v];
                        p[v] = u ;
                    }
        }
        for ( u=0 ; u<n ; u++) {
            for( v=0 ; v<n ; v++) {
                if(d[v] > d[u] + A[u][v] ) {
                    printf ("Negative edge ");
                    exit(0);
                }
            }
        }
}  // end of function
```

// cont.

OUTPUT

```
$./dv
Enter no. of vertices : 4
Enter Adjacency matrix :
 999  7   2  999
   1  999 999  5
   1  999 999  1
 999  2   1  999
```

999
Here indicates the absence of an edge

Router 0

0 cost 0

1 ← 3 ← 2 ← 0 cost 5

2 ← 0 cost 2

3 ← 2 ← 0 cost 3

Router 1

0 ← 1 cost 1

1 cost 0

2 ← 0 ← 1 cost 3

3 ← 2 ← 0 ← 1 cost 4

Router 2

0 ← 2 cost 1

1 ← 3 ← 2 cost 3

2 ← cost 0

3 ← 2 cost 1

Router 3

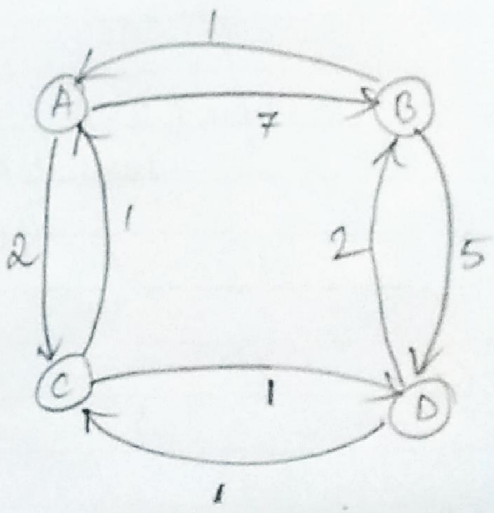
0 ← 2 ← 3 cost 2

1 ← 3 cost 2

2 ← 3 cost 1

3 cost 0

Graph



```
int main()
{
    printf("Enter the no. of vertices : "); scanf("%d", &n);
    printf("Enter adjacency matrix\n");
    for(int i=0; i<n; i++)
        for(int j=0; j<n; j++)
            scanf("%d", &A[i][j]);
    int source=0;
    for(source=0; source<n; source++)
    {
        for(int i=0; i<n; i++){
            d[i] = 999;
            p[i] = -1;      {
        d[source] = 0;

        printf("Router %d\n", source);

        for(int i=0; i<n; i++) {
            if(i != source) {
                int j=i;
                while(p[j] != -1){
                    printf("%d <- ", j);
                    j = p[j];
                }
            }
        printf("%d\t cost %d\n", source, d[i]);
        }
    }
    return 0; } // end of program.
```

Teacher's Signature : _____