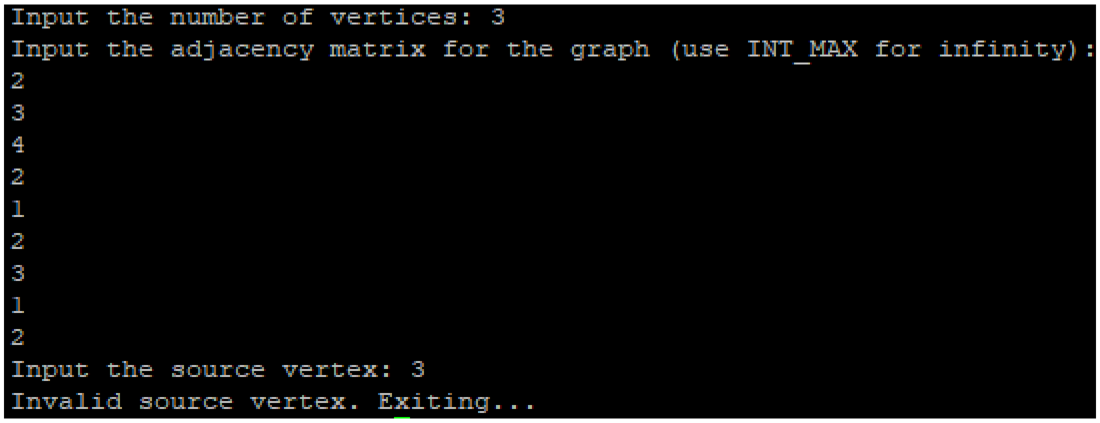
EX NO 14 DIJIKSTRA’S ALGORITHM

```
#include <stdio.h>
#define size 8
#define INFINITY 10000000;
int g[size][size]={ {0,2,6,0,0,0,0,0},
                   {2,0,0,2,6,0,0,0},
                   {6,0,0,1,0,0,4,0},
                   {0,2,1,0,0,2,0,0},
                   {0,6,0,0,0,3,0,1},
                   {0,0,0,2,3,0,2,0},
                   {0,0,0,2,0,2,0,2},
                   {0,0,0,0,1,0,2,0} };

struct vertex_info
{
    int length;
    int pred;
    char state;
}v[size];

int main()
{
            int i;
            for (i=0;i<size;i++)
            {
                        v[i].length=INFINITY;
                        v[i].pred=-1;
                        v[i].state='N';
            }
            int s=0;
            int d=7;
            v[s].length=0;
            v[s].state='V';

             do
            {
                        int i;
                        for(i=0;i<size;i++)
                        {
                                    if (g[s][i]!=0 &&v[i].state=='N')
```

```
                                {
                if(v[i].length>v[s].length+g[s][i])
                                {
                    v[i].length=g[s][i]+v[s].length;
                                          v[i].pred=s;

printf("\nlength[%d]=%d\tpred[%d]=%d",i,v[i].length,i,v[i].pred);
                                }
                                }
            }
            int min=INFINITY;
            s=0;
                      for(i=0;i<size;i++)
            {
                                if(v[i].state=='N'&& v[i].length<min)
                                {
                                min=v[i].length;
                                s=i;
                                }
                 }
            v[s].state='V';
            }while(s!=d);
    i=size;
    int path[size];
    printf("\n\nPath=%d->",s);
    do
    {
                path[i--]=s;
              s=v[s].pred;
              printf("%d->",s);
    }while(s>0);

}
```

OUTPUT:



```
Input the number of vertices: 3
Input the adjacency matrix for the graph (use INT_MAX for infinity):
2
3
4
2
1
2
3
1
2
Input the source vertex: 3
Invalid source vertex. Exiting...
```