```
// Display mode
mode(0);

// Display warning for floating point exception
ieee(1);

clc;
siz = input("Enter the size of the matrix --> ");
h = input("Enter the matrix --> ");
s = input("Enter the source vertex --> ");
d = input("Enter the destination vertex --> ");

for i =1:siz
  for j =1:siz
    if h(i,j) ==(-1) then
      h(i,j) = %inf;
    end;
  end;
end;

matriz_costo = h;
//function [sp, spcost] = dijkstra(matriz_costo, s, d)

n = siz;
S(1:n) = 0;//s, vector, set of visited vectors
dist(1:n) = %inf;// it stores the shortest distance between the source node and any other node;
prev(1:n) = (n+1);// Previous node, informs about the best previous node known to reach each  network
 node

dist(s) = (0);

while (sum(S) ~= (n))
  candidate = [];
  for i =1:n
    if S(i)==0 then
      candidate = [candidate,dist(i)];
    else
      candidate = [candidate,%inf];
    end;
  end;
  [u_index,u] = min(candidate);
  S(u) = 1;
  for i = 1:n
    if (dist(u)+(matriz_costo(u,i))) < dist(i) then
      dist(i) = dist(u)+(matriz_costo(u,i));
      prev(i) = u;
    end;
  end;
end;
sp = [d];
while (sp(1) ~= s )
  if (prev(sp(1)))<=n then
    sp = [prev(sp(1)) sp];
  else
    error;
  end;
end;
spcost = dist(d);
// L.56: No simple equivalent, so mtlb_fprintf() is called.
printf("\nThe least cost path from source node ""%d"" to destination node ""%d"" -->",s,d);
disp(sp);
// L.58: No simple equivalent, so mtlb_fprintf() is called.
printf("\nLeast cost is --> %g",spcost);
// L.59: No simple equivalent, so mtlb_fprintf() is called.
disp(" ------------ PROCESSING COMPLETED ---------------");
```