**PROGRAM FOR DIJKSTRA’S ALGORITHM:**

#include <stdio.h>

#include <stdlib.h>

//#define INF = 999;

int cost[30][30];

int arr[30][30];

int main()

{

int i,j,k,n;

int INF=999;

printf("\n Enter the total number of vertices:\n");

scanf("%d",&n);

printf("\n Enter the Adj. Matrix for cost of edges:\n");

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

{

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

{

printf("Enter:");

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

if(cost[i][j]==0)

{

arr[i][j] = INF;

}

else

{

arr[i][j] = cost[i][j];

}

}

printf("Next Row:");

}

printf("\n The Entered Matrix is:\n");

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

{

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

{

printf("%d\t|",cost[i][j]);

}

printf("\n");

}

printf("\n Adj. Matrix of cost of edges:\n");

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

{

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

{

printf("%d\t|",arr[i][j]);

}

printf("\n");

}

for(k=0;k<n;k++)

{

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

{

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

{

if(i==j && cost[i][j]==0)

{

arr[i][j] = 0;

}

else

{

if(arr[i][j] > (arr[i][k] + arr[k][j]))

{

arr[i][j] = (arr[i][k] + arr[k][j]);

}

}

}

}

}

printf("\n The final distance are:\n");

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

{

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

{

printf("%d\t|",arr[i][j]);

}

printf("\n");

}

int v;

printf("\nEnter the vertex no. from which the shortest path is to find:");

scanf("%d",&v);

for(i=v;;)

{

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

{

printf("\nDistance from %d to %d is: %d",i,j,arr[i][j]);

}

break;

}

return 0;

}

**OUTPUT**:

Enter the total number of vertices:

6

Enter the Adj. Matrix for cost of edges:

Enter:0

Enter:2

Enter:1

Enter:99

Enter:99

Enter:3

Next Row:Enter:2

Enter:0

Enter:99

Enter:4

Enter:99

Enter:2

Next Row:Enter:1

Enter:99

Enter:0

Enter:99

Enter:2

Enter:99

Next Row:Enter:99

Enter:4

Enter:99

Enter:0

Enter:5

Enter:3

Next Row:Enter:99

Enter:99

Enter:2

Enter:5

Enter:0

Enter:99

Next Row:Enter:3

Enter:2

Enter:99

Enter:3

Enter:99

Enter:0

Next Row:

The Entered Matrix is:

0 |2 |1 |99 |99 |3 |

2 |0 |99 |4 |99 |2 |

1 |99 |0 |99 |2 |99 |

99 |4 |99 |0 |5 |3 |

99 |99 |2 |5 |0 |99 |

3 |2 |99 |3 |99 |0 |

Adj. Matrix of cost of edges:

999 |2 |1 |99 |99 |3 |

2 |999 |99 |4 |99 |2 |

1 |99 |999 |99 |2 |99 |

99 |4 |99 |999 |5 |3 |

99 |99 |2 |5 |999 |99 |

3 |2 |99 |3 |99 |999 |

The final distance are:

0 |2 |1 |6 |3 |3 |

2 |0 |3 |4 |5 |2 |

1 |3 |0 |7 |2 |4 |

6 |4 |7 |0 |5 |3 |

3 |5 |2 |5 |0 |6 |

3 |2 |4 |3 |6 |0 |

Enter the vertex no. from which the shortest path is to find: 2

Distance from 2 to 0 is: 1

Distance from 2 to 1 is: 3

Distance from 2 to 2 is: 0

Distance from 2 to 3 is: 7

Distance from 2 to 4 is: 2

Distance from 2 to 5 is: 4