

LAB PROGRAM 3:- Write a program

distance vector algorithm to find path for transmission

Code:

```
#include <stdlib.h>
#define nul 1000
#define nodes 10
int no;
struct node
{
    int a[nodes][4];
} router[nodes];
void init (int r)
{
    int i;
    for (int i = 1; i <= no; i++)
    {
        router[r].a[i][1] = i;
        router[r].a[i][2] = 999;
        router[r].a[i][3] = nul;
    }
    router[r].a[r][2] = 0;
    router[r].a[r][3] = r;
}
void inp (int r)
```

```

int i;
printf("\nEnter dist from the node 'od'
to other nodes", r);
printf("\n pls enter 999 if there is
no direct route\n", r);
for (i = 1; i <= no; i++)
{
    if (i == r)
    {
        printf("\nEnter dist to the node 'od:' i)
scanf("%d", &router[r].a[i][2]);
router[r].a[i][3] = i;
    }
}
}

```

```

void display (int r)
{
    int i, j;
    printf("\n\nThe routing table for
node 'od' is as follows:", r);
    for (i = 1; i <= no; i++)
    {
        if (router[r].a[i][2] == 999)
            printf("\n |t|t|t. 'od' |t no link
|t no hop", router[r].a[i][1]);
        else
            printf("\n |t|t|t|t 'od' |t 'od' |t
|t d", router[r].a[i][1], router[r].
a[i][2], router[r].a[i][5]);
    }
}

```

int outside the control

it involves g
as a tool, kn
to a project
ade during
is critical
al sampl
to use a
ulation of

is flow
and a di
ring

```
void dsa_algo(int r)
```

```
{  
    int i, j, z;
```

```
    for (int i = 1; i <= n0; i++)
```

```
    {  
        if (router[r].a[i][2] != 0 &&  
            router[r].a[i][2] != 0)
```

```
        {
```

```
            for (j = 1; j <= n0; j++)
```

```
            {
```

```
                z = router[r].a[i][2] + router[r].a[j][2]
```

```
                if (z < 0)
```

```
                {  
                    if (router[r].a[j][2] > z)
```

```
                    {
```

```
                        router[r].a[j][2] = z;
```

```
                        router[r].a[j][3] = i;
```

```
                    }  
                }  
            }  
        }
```

```
int main ()
```

```
{
```

```
    int i, j, x, y;
```

```
    char choice;
```

```
    printf("Enter the no. of nodes  
    required (less than 100): ");
```

```
    scanf("%d", &n0);
```

```
    for (i = 1; i <= n0; i++)
```

```
    {  
        init(i);
```


inp (i),
3

printf ("In the configuration of the nodes
after initialization is as follows");
for (i = 1; i <= no; i++)
display (i);

for (i = 1; i <= no; i++)
dijkstra_algo (i);

printf ("In the configuration of the nodes
after computation of paths is as follows");
for (i = 1; i <= no; i++)
display (i);
while (1)

{
printf ("I wanna continue (y/n):");
scanf ("%c", &choice);
if (choice == 'n')
break;

printf ("Enter the nodes b/n
which shortest path is to be found: (n)");

scanf ("%d %d", &x, &y);

printf ("In the length of the shortest
path is %d", router [x] [y] [d]);

}

Output

Enter the no. of nodes

Distance from node 1 to other nodes

to node 1 : 2
to node 3 : 999
to node 4 : 1

Distance from node 2 to other nodes

to node 1 : 2
to node 3 : 3
to node 4 : 7

Distance from node 3 to other nodes

to node 1 : 999
to node 2 : 3
to node 4 : 11

Distance from node 4 to other nodes

to node 1 : 1
to node 2 : 7
to node 3 : 11

classmate
Date _____
Page _____

Routing table for node 1 as follows

1	0	
2	2	d
3	∞	d
4	1	no hop
		d

Routing table for node 2

1	2	
2	0	d
3	3	d
4	7	d

Routing table for node 3

1	∞	no hop
2	3	d
3	0	d
4	"	d

Routing table for node 4

1	1	d
2	7	d
3	"	d
4	0	d

91

Graph Considered

