

Program: All pair Shortest Path(Floyd-Warshall Algorithm) Niyati Savant

Code:

```
#include<stdio.h>

int max=100;

void printMatrix(int D[5][5],int n)
{
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
            printf("%d\t",D[i][j]);
        printf("\n");
    }
}

int main()
{
    printf("Niyati's Code");
    int i,j,k,n;
    int D[5][5];
    int P[5][5];
    printf("How many nodes ?");
    scanf("%d",&n);
    printf("Enter the elements \n");
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
            scanf("%d",&D[i][j]);
        printf("\n");
    }

    /*{ {0,3,8,100,-4},
        {100,0,100,1,7},
```

```

        {100,4,0,100,100},
        {2,100,-5,0,100},
{100,100,100,6,0} },*/

```

```

for(int i=0;i<n;i++)
{
    for(int j=0;j<n;j++)
    {
        if((i==j) || D[i][j]==max)
            P[i][j]=0;
        else
            P[i][j]=i;
    }
}

printf("Initially The Distance Matrix D is \n");
printMatrix(D,5);
printf("\n");
printf("And The Path Matrix P is \n");
printMatrix(P,5);

```

```

for(k=0;k<n;k++) //Intermediate Vertex
{
    for(i=0;i<n;i++) //row
    {
        for(j=0;j<n;j++)//column
        {
            if(D[i][j] > (D[i][k] +D[k][j]))
            {
                D[i][j]= D[i][k]+D[k][j];
                P[i][j]=P[k][j];
            }
        }
    }
}

```

```

        }
    }
}
printf("The Distance Matrix D%d is \n",k+1);
printMatrix(D,5);
printf("\n");
printf("The Path Matrix P%d is \n",k+1);
printMatrix(P,5);
}

```

```
return 0;
```

```
}
```

Output:

Niyati's Code

How many nodes ?5

Enter the elements

0 3 8 100 -4

100 0 100 1 7

100 4 0 100 100

2 100 -5 0 100

100 100 100 6 0

Initially The Distance Matrix D is

0	3	8	100	-4
100	0	100	1	7
100	4	0	100	100
2	100	-5	0	100
100	100	100	6	0

And The Path Matrix P is

0	0	0	0	0
0	0	0	1	1
0	2	0	0	0
3	0	3	0	0
0	0	0	4	0

The Distance Matrix D1 is

0	3	8	100	-4
100	0	100	1	7
100	4	0	100	96
2	5	-5	0	-2
100	100	100	6	0

The Path Matrix P1 is

0	0	0	0	0
0	0	0	1	1
0	2	0	0	0
3	0	3	0	0
0	0	0	4	0

The Distance Matrix D2 is

0	3	8	4	-4
100	0	100	1	7
100	4	0	5	11
2	5	-5	0	-2
100	100	100	6	0

The Path Matrix P2 is

0	0	0	1	0
0	0	0	1	1
0	2	0	1	1
3	0	3	0	0
0	0	0	4	0

The Distance Matrix D3 is

0	3	8	4	-4
100	0	100	1	7
100	4	0	5	11
2	-1	-5	0	-2
100	100	100	6	0

The Path Matrix P3 is

0	0	0	1	0
0	0	0	1	1
0	2	0	1	1
3	2	3	0	0
0	0	0	4	0

The Distance Matrix D4 is

0	3	-1	4	-4
3	0	-4	1	-1
7	4	0	5	3
2	-1	-5	0	-2
8	5	1	6	0

The Path Matrix P4 is

0	0	3	1	0
3	0	-4	1	-1
7	4	0	5	3
2	-1	-5	0	-2
8	5	1	6	0

The Distance Matrix D5 is

0	1	-3	2	-4
3	0	-4	1	-1
7	4	0	5	3
2	-1	-5	0	-2
8	5	1	6	0

The Path Matrix P5 is

0	2	3	4	0
3	0	3	1	0
3	2	0	1	0
3	2	3	0	0
3	2	3	4	0