

MUSKAN GUPTA
IBM19CS091

ADA LAB TEST 2

Q Implement all pair shortest paths problem using Floyd's algorithm

Modification :- Shortest path between the given source and destination

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void floyds();
```

```
int min(int, int);
```

```
int c[10][10], d[10][10], i, j, k, n, src, dest;
```

```
void main()
```

```
{
```

```
    printf("Enter number of vertices\n");
```

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

```
    printf("Enter cost adjacency matrix\n");
```

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

```
    {
```

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

```
        {
```

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

```
        }
```

```
    }
```

```
    floyds();
```

①

Muskan

MUSKAN GUPTA
IBM19C3091

```
printf (" \n Shortest path between every pair of vertices \n");  
for (i=1; i <= n; i++)  
{  
    for (j=1; j <= n; j++)  
    {  
        printf (" %.d ", d[i][j]);  
    }  
    printf (" \n");  
}  
  
printf (" \n Enter the source vertex :");  
scanf (" %.d ", &src);  
printf (" Enter the destination vertex: \n");  
scanf (" %.d ", &dest);  
  
printf (" Shortest path between source vertex %.d and  
destination vertex %.d = %.d", src, dest, dest  
d[src][dest]);  
  
getch();  
}  
  
int min ( int a, int b)  
{  
    if (a < b)  
    {  
        return (a);  
    }  
    else  
        return (b);  
}
```

(2)

Muska

MUSKAN GUPTA
IBM19CS091

void floyd()

```
{  
  for (i=1; i <= n; i++)  
  {
```

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

```
    {  
      d[i][j] = c[i][j];
```

```
    }
```

```
  }
```

```
  for (k=1; k <= n; k++)
```

```
  {
```

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

```
    {
```

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

```
      {
```

```
        d[i][j] = min(d[i][j], d[i][k] + d[k][j]);
```

```
      }
```

```
    }
```

```
  }
```

```
}
```