

Sakshi Srivastava

IBM18C5090-SB-

B3 Batch.

Implement Dijkstra's algorithm to compute the shortest path through a graph.

```
int dijkstra (int cost[][N], int  
source, int target);
```

```
int dijkstra (int cost[][N], int  
source, target)
```

```
{ int dist[N], prev[N],  
  selected[N] = {0}, i, m, mini,  
  start, d, j;  
  char path[N];
```

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

```
  { dist[i] = 1N;  
    prev[i] = -1;
```

```
    start = source;  
    selected[start] = 1;
```

```
    dist[start] = 0;
```

```
    while (selected[target] == 0)
```

```
    { mini = 1N;
```

```
      m = 0;
```

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

```
      { d = dist[start] + cost[start][i];  
        if (d < dist[i] && selected[i] == 0)
```

```
        { dist[i] = d;  
          prev[i] = start;
```

```
        }
```

```
if (min > dist[i] && selected[i] == 0)
```

```
{ min = dist[i];  
  m = i;
```

```
}
```

```
start = m;
```

```
selected[start] = 1;
```

```
start = target;
```

```
j = 0;
```

```
while (start != 1)
```

```
{ path[j++] = start + 65;  
  start = prev[start];
```

```
path[j] = '\0';
```

```
strrev(path);
```

```
printf("%s", path);
```

```
return dist[target];
```

```
int main()
```

```
{ int cost[N][N], i, j, w, ch, co;
```

```
int source, target, n, y;
```

```
printf("It Shortest Path Algorithm (DIJKSTRA'S  
ALGORITHM) \n \n");
```

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

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

```
cost[i][j] = 1000;
```

```
for (x = 1; x < N; x++)
```

```
for (y = x + 1; y < N; y++)
```

```
{ printf("Enter the wt. of the path  
b/w node %d & %d \n", x, y);
```

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



```

cost[x][y] = cost[y][x] = w;
}
printf("%i\n", n);
}
printf("%i\n", n); // Enter the source
scanf("%i", &source);

printf("%i\n", n); // Enter the Target
scanf("%i", &target);
co = dijkstra(cost, source, target);
printf("%i\n", co); // Shortest Path
}

```