

**D.Y PATIL INTERNATIONAL UNIVERSITY**  
**School of Computer Science, Engineering and Applications**  
**Academic Year 2023-2024(monsoon semester)**  
**Practical Assignment- 07**  
**Subject: Design and Analysis of algorithm**

**Name:** Suryakant Upadhyay

**PRN:** 20220802043

**Class:** S.Y. B.Tech. SEM III

---

**Topic:** Dynamic Programming

- 1) Write a C code to implement Travelling salesman problem using dynamic programming method.

**CODE:**

```
#include <stdio.h>

int ary[10][10], completed[10], n, cost = 0;

void takeInput()
{
    int i, j;
    printf("Enter the number of villages: ");
    scanf("%d", &n);
    printf("\nEnter the Cost Matrix\n");
    for (i = 0; i < n; i++)
    {
        printf("\nEnter Elements of Row: %d\n", i + 1);
        for (j = 0; j < n; j++)
            scanf("%d", &ary[i][j]);
        completed[i] = 0;
    }

    printf("\n\nThe cost list is:");

    for (i = 0; i < n; i++)
    {
        printf("\n");
        for (j = 0; j < n; j++)
            printf("\t%d", ary[i][j]);
    }
}

int least(int c)
{
    int i, nc = 999;
    int min = 999, kmin;
```

```

for (i = 0; i < n; i++)
{
    if ((ary[c][i] != 0) && (completed[i] == 0))
    {
        if (ary[c][i] < min)
        {
            min = ary[c][i];
            kmin = ary[c][i];
            nc = i;
        }
    }
}
if (min != 999)
    cost += kmin;
return nc;
}

void mincost(int city)
{
    int i, ncity;
    completed[city] = 1;
    printf("%d--->", city + 1);
    ncity = least(city);
    if (ncity == 999)
    {
        ncity = 0;
        printf("%d", ncity + 1);
        cost += ary[city][ncity];
        return;
    }
    mincost(ncity);
}
int main()
{
    takeInput();
    printf("\n\nThe Path is:\n");
    mincost(0);
    printf("\n\nMinimum cost is %d\n", cost);
    return 0;
}

```

**OUTPUT:**

*Enter the number of villages: 4*

*Enter the Cost Matrix*

*Enter Elements of Row: 1*

*0 4 1 3*

*Enter Elements of Row: 2*

*4 0 2 1*

*Enter Elements of Row: 3*

*1 2 0 5*

*Enter Elements of Row: 4*

*3 1 5 0*

*The cost list is:*

*0 4 1 3*

*4 0 2 1*

*1 2 0 5*

*3 1 5 0*

*The Path is:*

*1->3->2->4->1*

*Minimum cost is 7*