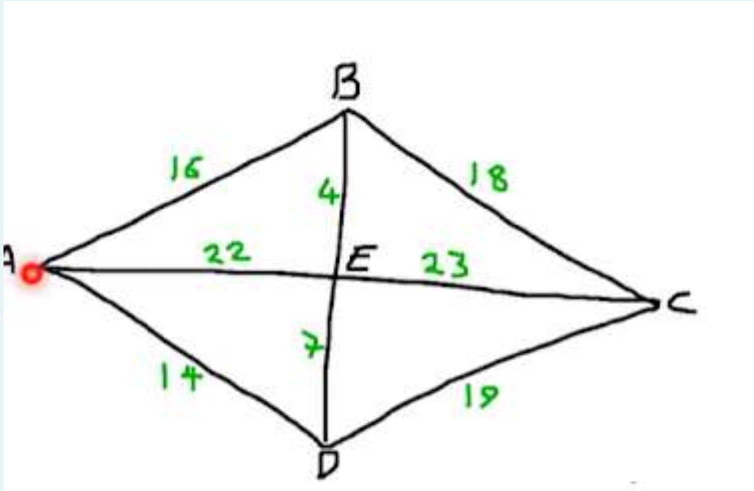
# Experiment No. 3.3

**Student Name: Rishav Kumar** **UID: 22MCC20039**
**Branch: MCA - CCD** **Section/Group: MCD-1/ Grp B**
**Semester: I** **Date of Performance: 05th Jan 22**
**Subject Name: DAA Lab** **Subject Code: 22CAP-646**

## 1. Aim/Overview of the practical:

Implement any scheme to find the optimal solution for the Traveling Salesperson problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.



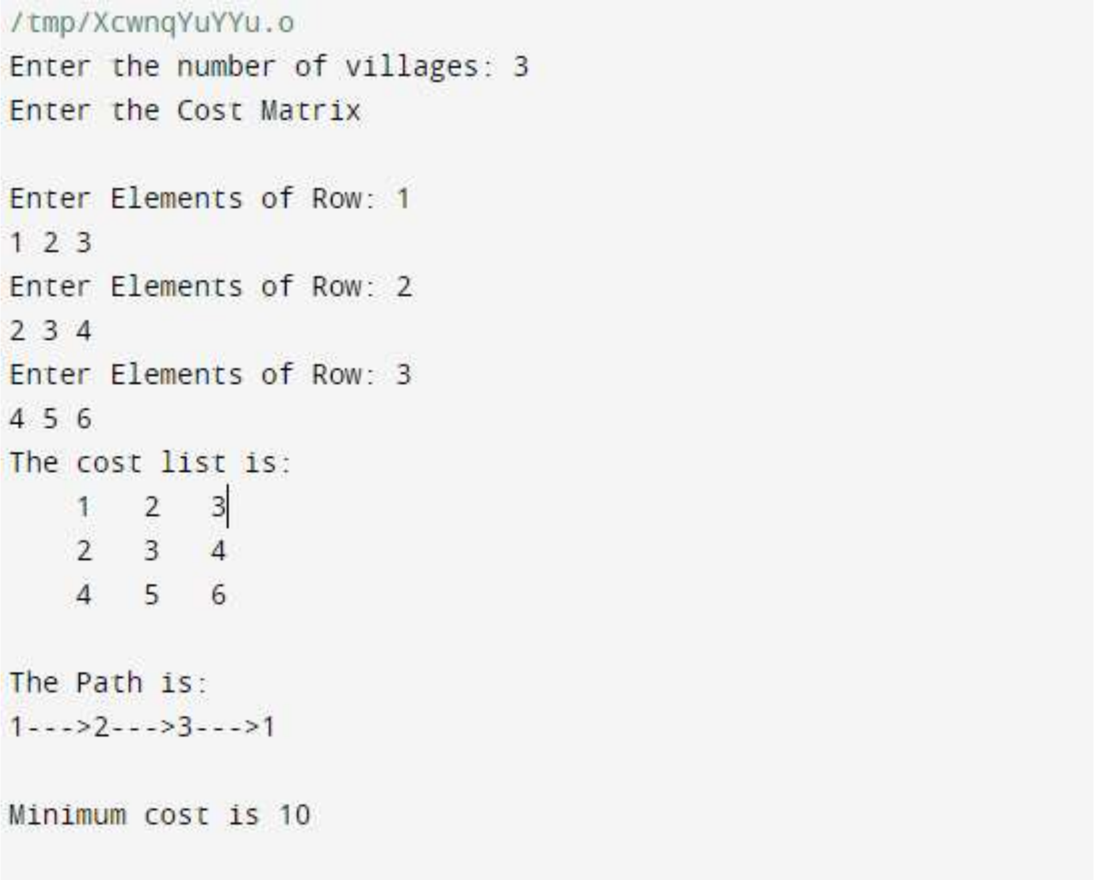
## 2. Code for practical:

```
#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]);
  }
}

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 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] + ary[i][c] < min) {
        min = ary[i][0] + ary[c][i];
        kmin = ary[c][i];
        nc = i;
      }
  }

  if (min != 999)
    cost += kmin;

  return nc;
}

int main() {
  takeInput();

  printf("\n\nThe Path is:\n");
  mincost(0); //passing 0 because starting vertex

  printf("\n\nMinimum cost is %d\n ", cost);
```

```
    return 0;
}
```

### 3. Output:



```
/tmp/XcwnqYuYYu.o
Enter the number of villages: 3
Enter the Cost Matrix

Enter Elements of Row: 1
1 2 3
Enter Elements of Row: 2
2 3 4
Enter Elements of Row: 3
4 5 6
The cost list is:
    1   2   3
    2   3   4
    4   5   6

The Path is:
1--->2--->3--->1

Minimum cost is 10
```

**************************************** THE END ************************************************