10. Write Java programs to

(b) Implement **Travelling Sales Person problem** using Dynamic programming.

```
import java.util.Scanner;
public class Tsp

{
   public static void main(String[] args)
   {
      Scanner in = new Scanner(System.in);
      int c[][]=new int[10][10], tour[]=new int[10];
      int i, j,cost;
      System.out.print("Enter No. of Cities: ");
      int n = in.nextInt();
      if(n==1)
      {
         System.out.println("Path is not possible!");
         System.exit(0);
      }

      System.out.println("Enter the Cost Matrix:");
      for(i=1;i<=n;i++)
         for(j=1;j<=n;j++)
            c[i][j] = in.nextInt();
      for(i=1;i<=n;i++)
         tour[i]=i;

      cost = tspdp(c, tour, 1, n);

      System.out.print("The Optimal Tour is: ");
      for(i=1;i<=n;i++)
         System.out.print(tour[i]+"->");
      System.out.println("1");
      System.out.println("Minimum Cost: "+cost);
   }


   static int tspdp(int c[][], int tour[], int start, int n)
   {
      int mintour[]=new int[10], temp[]=new int[10], mincost=999,ccost, i, j, k;
```

```
        if(start == n-1)
        {
            return (c[tour[n-1]][tour[n]] + c[tour[n]][1]);
        }
        for(i=start+1; i<=n; i++)
        {
            for(j=1; j<=n; j++)
                temp[j] = tour[j];
            temp[start+1] = tour[i];
            temp[i] = tour[start+1];
            if((c[tour[start]][tour[i]]+(ccost=tspdp(c,temp,start+1,n)))<mincost)
            {
                mincost = c[tour[start]][tour[i]] + ccost;
                for(k=1; k<=n; k++)
                    mintour[k] = temp[k];
            }
        }
        for(i=1; i<=n; i++)
            tour[i] = mintour[i];
        return mincost;
    }
}
```