```
package places;
import java.util.*;
public class project
{
	String[] places;
	Hashtable <String, String> state;
	public int vertices;
	public project()
	{}
	public project(int a)
	{
		vertices=a;
		places=new String[vertices];
		state=new Hashtable<String, String>();
		places[0]="Banglore";
		places[1]="Vishakapattanam";
		places[2]="Chennai";
		places[3]="Mumbai";
		places[4]="Goa";
		places[5]="Manglore";
		places[6]="Mysore";
		places[7]="Agra";
		places[8]="Trivendram";
		places[9]="Kolkatta";
		places[10]="Srinagar";
		places[11]="Odisha";
		places[12]="Chandigarh";
		places[13]="Kota";
		places[14]="Patna";
		places[15]="Hyderabad";
		places[16]="Ranchi";
		places[17]="Gangtok";
		places[18]="Gandhinagar";
		places[19]="Ahmedabad";
		places[20]="Surat";
		places[21]="Jaipur";
		places[22]="Lucknow";
		places[23]="Nagpur";
		places[24]="Varanasi";


		state.put("Karnataka","Banglore, Mysore and Manglore");
		state.put("Rajasthan","Kota and Jaipur");
		state.put("West Bengal","Kolkatta");
		state.put("Andra Pradesh","Vishakapattanam and Hyderabad");
		state.put("Tamil Nadu","Chennai");
		state.put("Gujarat","Surat, Gandhi Nagar and Ahmadabad");
		state.put("Uttar Pradesh","Lucknow, Varanasi and Agra");
		state.put("Maharashtra","Mumbai and Nagpur");
		state.put("Kerala","Trivendram");
		state.put("Jammu & Kashmir","Srinagar");
```

```
		state.put("Orissa","Odisha");
		state.put("Jharkhand","Ranchi");
		state.put("Chandigarh","Chandigarh");
		state.put("Sikkim","Gangtok");
		state.put("Goa","Goa");
		state.put("Bihar","Patna");
	}
	public String retpla(int i)
	{
		return places[i];
	}
	public int retind(String name)
	{

		int flag=0,i;
		for(i=0;i<vertices;i++)
		{
			if(name.equals(places[i]))
			{
				flag=1;
				break;
			}
		}
		if(flag==1)
			return i;
		else
			return -1;
	}
	public int retvert()
	{
		return vertices;
	}
	public String retstate(String name)
	{
		String stat="";
		stat=stat+state.get(name);
		return stat;
	}

}


package time;
import java.util.*;
import kruskal.*;
import places.*;
public class project1 extends kruskal
{
	 int[][] time;
	int vertices;
	public project1()
	{}
```

```
	public project1(int vertices)
	{
		this.vertices=vertices;
		time=new int[vertices][vertices];
	}
	public void set()
	{
		for(int i=0;i<vertices;i++)
		{
			for(int j=0;j<vertices;j++)
			{
				if(i==j)
					time[i][j]=99999;
				else
				{
					Random r=new Random();
					time[i][j]= r.nextInt((360 - 60)+1)+60;
				}
			}
		}
	}
	public int[][] rettime()
	{
		return time;
	}


}

package money;
import kruskal.*;
import java.util.*;
public class project2 extends kruskal
{
	int[][] money;
	int vertices;
	public project2()
	{}
	public project2(int vertices)
	{
		this.vertices=vertices;
		money=new int[vertices][vertices];
	}
	public void set()
	{
		Calendar cal=Calendar.getInstance();
		Random r=new Random();
		for(int i=0;i<vertices;i++)
		{
			for(int j=0;j<vertices;j++)
			{
				if(i==j)
```

```
                                        money[i][j]=99999;
                                else
                                        money[i][j]=r.nextInt((6000-3000)+1)+3000;
                        }
                }
                for(int i=0;i<vertices;i++)
                {
                        for(int j=0;j<vertices;j++)
                        {
                                if((cal.get(Calendar.MONTH)>=3 )&&
(cal.get(Calendar.MONTH)<=5))
                                        money[i][j]=(int)(money[i][j]*1.2);
                                else if((cal.get(Calendar.MONTH)>=6 )&&
(cal.get(Calendar.MONTH)<=8))
                                        money[i][j]=(int)(money[i][j]*1.5);
                                else if((cal.get(Calendar.MONTH)>=9 )&&
(cal.get(Calendar.MONTH)<=11))
                                        money[i][j]=money[i][j]*2;
                        }
                }
        }
        public int[][] retmoney()
        {
                return money;
        }

}


package finall;
import places.*;
import time.*;
import money.*;
import java.util.*;
import kruskal.*;
import floyd.*;
public class project3
{
        String source;
        String destination;
        project3()
        {}
        project3(String n1,String n2)
        {
                source = n1;
                destination = n2;
        }
        public static void main(String args[])
        {
                Scanner s=new Scanner(System.in);
                project obj=new project(25);
                System.out.println("Enter the state: ");
```

```
		String state=s.nextLine();
		System.out.print("Airports in "+state+" : ");
		System.out.println(obj.retstate(state));
		System.out.println("Enter all for an All India Tour and single for a journey from ur place to a different place ... ");
		String p=s.nextLine();
		if(p.equals("all"))
		{
			System.out.println("Enter the boarding point: ");
			String source=s.nextLine();
			int i=obj.retind(source);
			project1 p1=new project1(obj.retvert());
			p1.set();
			project2 p2=new project2(obj.retvert());
			p2.set();
			kruskal k=new kruskal(p1.rettime(),obj.retvert());
			kruskal kk=new kruskal(p2.retmoney(),obj.retvert());
			int mintime=2*(k.kruskals(i,i,0));
			int mincost=2*(kk.kruskals(i,i,0));
			int hr1=mintime/60;
			int min1=mintime%60;
			System.out.print("All India tour takes atleast "+mincost+" Rs if you book today...");
			if(min1!=0)
				System.out.println(" and it is a "+hr1+" hour "+min1+" minutes toor...");
			else
				System.out.println(" and it is a "+hr1+" hours toor... ");

		}
		else if(p.equals("single"))
		{
			System.out.println("Enter the boarding point: ");
			String source=s.nextLine();
			System.out.println("Enter the dropping point: ");
			String desti=s.nextLine();
			System.out.println("Enter time if u prefer a fast journey and money if u prefer a cheaper journey...");
			String opt=s.nextLine();
			project3 obj1=new project3(source,desti);
			int i=obj.retind(obj1.source);
			int j=obj.retind(obj1.destination);
			if(opt.equals("time"))
			{
				project1 t1=new project1(obj.retvert());
				t1.set();
				kruskal k1=new kruskal(t1.rettime(),obj.vertices);
				int time1=k1.kruskals(i,j,1);
				int time2=floyd.floyds(t1.rettime(),i,j,obj.vertices);
				int time=floyd.min(time1,time2);
				int hr=time/60;
				int min=time%60;
```

```
                                if(min!=0)
                                        System.out.println("Faster journey from "+obj1.source+" to
"+obj1.destination+" takes "+time+" minutes.. i.e, "+hr+" hours and "+min+" minutes ...");
                                else
                                        System.out.println("Faster journey from "+obj1.source+" to
"+obj1.destination+" takes "+time+" minutes.. i.e, "+hr+" hours... ");
                        }
                        else if(opt.equals("money"))
                        {
                                project2 t2=new project2(obj.retvert());
                                t2.set();
                                kruskal k2=new kruskal(t2.retmoney(),obj.vertices);
                                int cost1=k2.kruskals(i,j,1);
                                int cost2=t2.floyds(t2.retmoney(),i,j,obj.vertices);
                                int cost=floyd.min(cost1,cost2);
                                System.out.println("Cheper journey from "+obj1.source+" to
"+obj1.destination +"takes "+cost+" Rs...");
                        }
                }
        }
}


package kruskal;
import floyd.*;
public class kruskal extends floyd
{
        int[][] input;
public int vertices;
public int[][] output;

        public kruskal()
        {}
        public kruskal(int[][] input,int ver)
        {
                this.input=new int[vertices][vertices];
                this.input=input;
                vertices=ver;
                output=new int[vertices][vertices];
                for(int i=0;i<vertices;i++)
                {
                        for(int j=0;j<vertices;j++)
                                output[i][j]=99999;
                }
        }

        public int retpar(int parent[],int u)
        {
                if(parent[u]==u)
                        return u;
                return retpar(parent,parent[u]);
        }
```

```
	public int kruskals(int a,int b,int flag)
	{
		int parent[]=new int[vertices];
		int count=0,cost=0,mincost=0;
		for(int i=0;i<vertices;i++)
			parent[i]=i;
		while(count!=vertices-1)
		{
			for(int i=0;i<vertices;i++)
			{
				for(int j=0;j<vertices;j++)
				{
					int u=retpar(parent,i);
					int v=retpar(parent,j);
					if(u!=v)
					{
						output[count][j]=input[i][j];
						count+=1;
						parent[j]=i;
						mincost=mincost+input[i][j];
					}
				}
			}
		}
		if(flag==1)
		{
			cost=floyd.floyds(output,a,b,vertices);
			return cost;
		}
		return mincost;
	}
}


package floyd;
public class floyd
{
	public static int min(int a,int b)
	{
		if(a<b)
			return a;
		return b;
	}

	public static int floyds(int graph[][],int a,int b,int vertices)
	{
		for(int k=0;k<vertices;k++)
		{
			for(int j=0;j<vertices;j++)
			{
				for(int i=0;i<vertices-1;i++)
```

```
                                        graph[i][j]=min(graph[i][j],(graph[i][k]+graph[k][j]));
                                }
                        }
                        return graph[a][b];
                }
        }
```