

PROJECT 1

'ELECTRIC CAR TRAVELER'

SUBJECT NAME= Advanced Algorithm

SUBJECT CODE= CPSC 535

INSTRUCTOR= Prof. Doina Bein

GROUP MEMBERS

Member 1= Lency Lakhani

Email Id= lencylakhani@csu.fullerton.edu

CWID ID= 885196055

Member 2= Hetal Patel






Email Id = hetal-patel.1994@csu.fullerton.edu

CWID ID= 885868455



Department of Computer Science
California State University, Fullerton

10 lines (6 sloc) | 672 Bytes

<>  Raw Blame    

535-project-1

Electric Car Traveler

Description: The project is to design an algorithm which computes the list of stops an electric car requires to recharge from the source to destination. If car has minimum charging then car should stop at charging station for recharge and if in case charging station is broken then car should have at least sufficient charging to travel back to the previous city for recharge. The program takes input for Capacity, number of cities and the distance between the cities and calculates the required stops the car needs to take.

Group members:

Hetal Patel hetal-patel.1994@csu.fullerton.edu

Lency Lakhani lencylakhani@csu.fullerton.edu

SUMMARY

We have developed a specific algorithm to implement 'Electric Car Traveler'. In this project, there are n number of cities such as ('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H') and n-1 distances between cities. C is mile range of electric car which is 300. Car starts from city A goes to the city H which takes some miles to reach to the destination. so, we have to design an algorithm if car has minimum charging so, we need to recharge it immediately at nearby charging station. But if, near by charging station is broken then car need to goes back to the previous station to recharge so, there should be sufficient charging in car to come back at previous station. L is a output list which, contains charging station where car has to stop for recharge.

PSEUDOCODE

step-1 take input for capacity in number of miles
step-2 take input for number of cities
step-3 take input for distance between the cities
step-4 take sum, previousdistance as zero and count as one
step-5 iterate Map using entryset

```
for (Map.Entry<Character, Integer> dist : maparr.entrySet()) {  
    valueArr2.add(dist.getKey());  
  
    if (count == 1) {  
        path.add(dist.getKey());  
    }  
  
    sum = sum + dist.getValue();  
    previousdistance = sum + dist.getValue();  
  
    if (previousdistance >= capacityc) {  
        sum = 0;  
        sum = sum + dist.getValue();  
        previousdistance = sum + dist.getValue();  
  
        path.add(valueArr2.get(valueArr2.size() - 2));  
    }  
  
    if (count == maparr.size()) {  
        path.add(dist.getKey());  
    }  
    count++;  
}
```

step-6 repeat step-5 until destination city.

step-7 return path as output list which contains number of stops for cars to recharge.

HOW TO RUN CODE

Method-1 Using Eclipse or other IDE

STEP 1: Download and open the files in IDE

STEP 2: Execute the code using run command

STEP 3: Go to the console window

STEP 4: Enter Maximum number of miles

STEP 5: Enter Number of cities

STEP 6: Enter distances from city A to H

STEP 7: At the end, console prints output List with cities where car has to stop for recharge

Method-2 Using Command Prompt/Terminal

STEP 1: Download the files

STEP 2: Open command prompt and go to the location of files

STEP 3: Compile the code using `javac Filename.java`

STEP 4: Run the file using `java Filename`

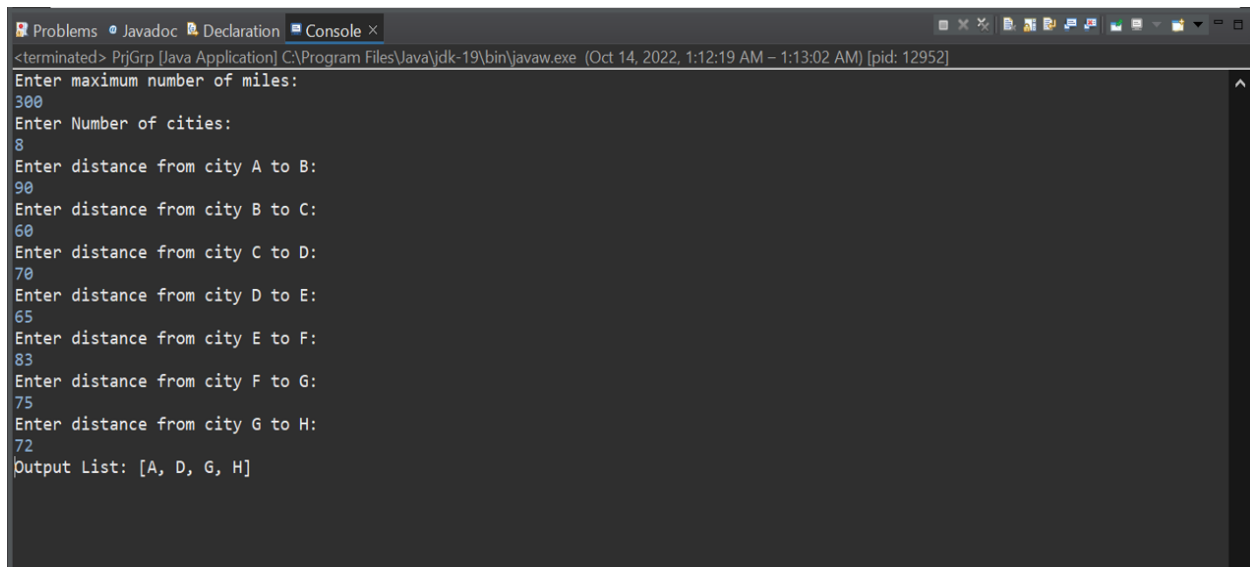
STEP 5: Enter Maximum number of miles

STEP 6: Enter Number of cities

STEP 7: Enter distances from city A to H

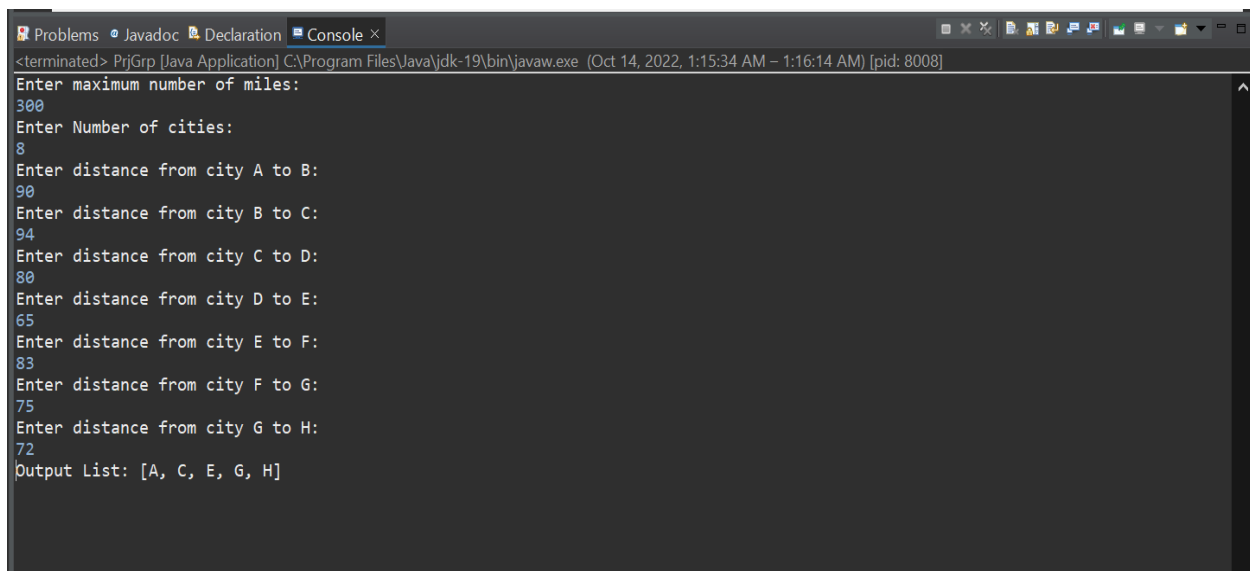
STEP 8: At the end, console prints output List with cities where car has to stop for recharge

EXAMPLE 1:



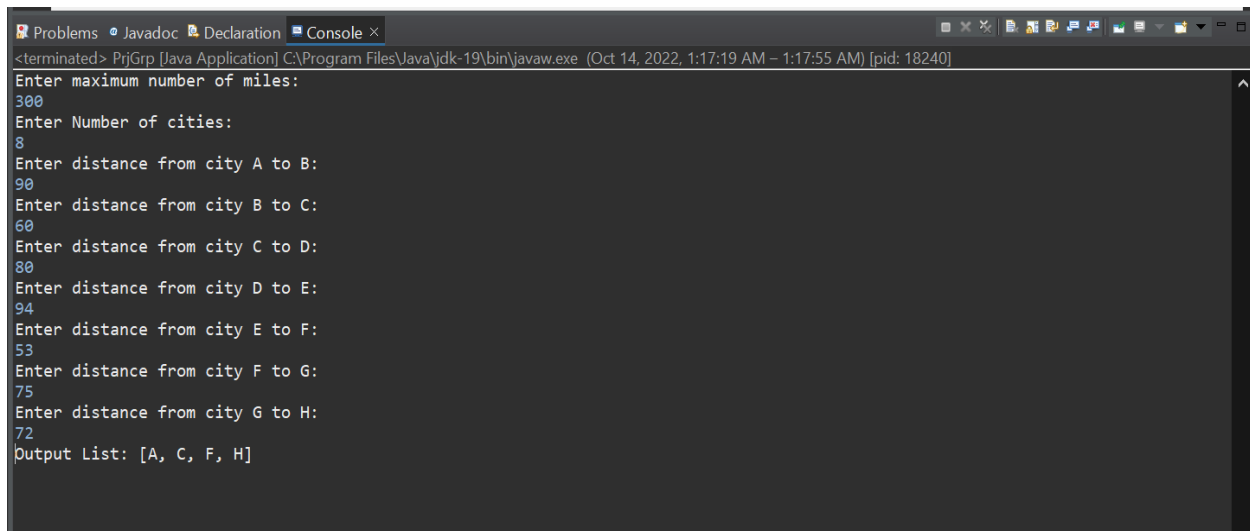
```
<terminated> PrjGrp [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Oct 14, 2022, 1:12:19 AM – 1:13:02 AM) [pid: 12952]
Enter maximum number of miles:
300
Enter Number of cities:
8
Enter distance from city A to B:
90
Enter distance from city B to C:
60
Enter distance from city C to D:
70
Enter distance from city D to E:
65
Enter distance from city E to F:
83
Enter distance from city F to G:
75
Enter distance from city G to H:
72
Output List: [A, D, G, H]
```

EXAMPLE 2:



```
<terminated> PrjGrp [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Oct 14, 2022, 1:15:34 AM – 1:16:14 AM) [pid: 8008]
Enter maximum number of miles:
300
Enter Number of cities:
8
Enter distance from city A to B:
90
Enter distance from city B to C:
94
Enter distance from city C to D:
80
Enter distance from city D to E:
65
Enter distance from city E to F:
83
Enter distance from city F to G:
75
Enter distance from city G to H:
72
Output List: [A, C, E, G, H]
```

EXAMPLE 3:



```
<terminated> PrjGrp [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Oct 14, 2022, 1:17:19 AM – 1:17:55 AM) [pid: 18240]
Enter maximum number of miles:
300
Enter Number of cities:
8
Enter distance from city A to B:
90
Enter distance from city B to C:
60
Enter distance from city C to D:
80
Enter distance from city D to E:
94
Enter distance from city E to F:
53
Enter distance from city F to G:
75
Enter distance from city G to H:
72
Output List: [A, C, F, H]
```