

Q7 : Hyperloop Passenger Booking

Problem :

You are to program a passenger booking system for a hyperloop transport of a particular station.

Assumptions

1. All the hyperloop routes with distance and the starting station will be given as input. *(See the below input section for details)*
2. The speed of the pod is constant but the distance between routes will vary.
3. A pod can travel from one connection to another in any direction in the given route. (i.e both A to B and B to A)
4. **One pod can accommodate only one passenger at a time.**
5. Passengers will be booked into their pods one by one.
 - a. All passengers will start from the given starting station.
 - b. Whenever a pod is started, the oldest person in the queue will board the pod first.
6. Passengers can arrive at any time as well as pods can start at any time.
7. Assume an infinite supply of pods and collisions will never happen.

System Logic :

Whenever a pod is started, the pod should pickup

- the oldest person by age from the passenger queue and
- take the fastest route

The command line should handle the following Commands

INIT

ADD_PASSENGER

START_POD

PRINT_Q

The arguments and details of the commands are mentioned below.

1. **INIT** Command - Initializes the system with
 - a. Number of interconnecting routes (**N**) and the Starting station.
 - b. Next **N** lines denotes connection between two interconnections and their distance.

Without initializing - all other commands should throw proper error.
2. **ADD_PASSENGER** command adds passengers to the line.
 - a. **ADD_PASSENGER X** adds **X** number of passengers to the line. **X** lines following the **ADD** command denote the passenger's name, age and destination
 - b. NAME AGE DEST
3. **START_POD** command starts pod with a passenger having highest age to his destination following the minimum interconnection points. Print the passenger name and route.
 - a. **START_POD X** starts **X** number of passengers of oldest age. (**X** lines are printed with name and route)
 - b. NAME ROUTE
4. **PRINT_Q** command prints the number of passengers and their details who are remaining in the queue.
 - a. COUNT
 - b. NAME AGE

EXAMPLE

Input:	Output:
INIT 7 A A B 3 A C 7 B D 2 B C 2 B E 5 C E 1 D E 3	
ADD_PASSENGER 1 RAVI 22 C	
ADD_PASSENGER 2 HARI 33 D BALA 10 E	
START_POD 1	<i>HARI A B D</i>
PRINT_Q	2 <i>RAVI 22</i> <i>BALA 10</i>
ADD_PASSENGER 1 KATHIR 22 B	
PRINT_Q	3 <i>RAVI 22</i> <i>BALA 10</i> <i>KATHIR 22</i>
START_POD 2	<i>RAVI A B C</i> <i>KATHIR A B</i>
START_POD 1	<i>BALA A B E</i>