Previous | Tourists Pairing | ShareDrive | Car Manufacturing Design Problem | MBS The Line | Next

ShareDrive

ShareDrive, a new ride sharing service, has recently started its operations in Pakistan. To attract more drivers to their app, they are offering a joining package for drivers. Some of the key features in offer include:

- First month is completely free for drivers
- 10% commission on earnings for the second month i.e if driver is earning 1000 Rs in second month, he will pay 100 rupees to Sharedrive.
- and 20% commission on next months i.e if driver is earning 1000 Rs in next month, he will pay 200 rupees to Sharedrive.

In order to pay back the commission amount to Sharedrive, drivers can pay back any amount via Easypaisa or Jazzcash. All the payments are available in Payments-API below. If the amount paid is more than the required commission amount, It will become part of their balance and can be adjusted later.

ShareDrive requires your help in calculating the total amount that a driver owes or needs to be paid by the company in 2022. You are provided with 2 API endpoints for all the data you need:

Rides API: https://www.jsonkeeper.com/b/DM5F
This API will return all the trips in JSON format and each trip contains driver id , fare and trip date that can help you calculate the required commission amount.

Date format is MM/DD/YYYY

Note: For test cases, please pick trips/rides after the joining date and ignore others.

Format: JSON. List of trip objects, one object looks like this

```
{
  "driver_id": 10006,
  "driver_name": "Herman Lowe",
  "trip_rating": 5,
  "trip_details": {
    "distance_km": 4,
    "time_taken": 17,
    "fare": 68,
    "customer_id": "0271707666"
  },
  "trip_date": "05/01/2022",
  "vehicle": {
    "model": "Galant",
    "number": "1G4CU541034935075"
  }
}
```

Payments API: https://www.jsonkeeper.com/b/9QRZ
This API will return all the payments made by the drivers to Sharedrive. Driver id and amount value in each record can help you calculate the total amount that has been paid back to Sharedrive.
Date format is MM/DD/YYYY

Note: For test cases, please ignore joining date for this API.

Format: JSON. List of payment objects, one object looks like this

```
{
  "driver_id": 10007,
  "driver_name": "Donnie Saunders",
```



```
}
```

Limitations (HINT):

- Package will start from 1st of that month, no matter when you join in that month. E-g: Zubair joins on 1st of oct and Ali joins on 20th of Oct, first month will expire on on 31-Oct 11:59 for both drivers
- All the dates will be in 2022. You can just check the month part instead of going for a full date comparison.
- Date format is MM/DD/YYYY

Input Format

The input will be read from a file. The filename/path will be passed to your program as the first command-line argument. First line of input will be an integer T representing the number of test cases. Each test case contains a comma separated driver ID and Joining date of the driver.

Output Format

Output will be the amount that needs to be paid by Driver or by Company. Amount driver needs to pay will be negative. Format your output to fixed 1 decimal places.

Sample Test Case Input:

```
2
10017, 04/01/2022
10011, 01/01/2022
```

Sample Test Case Output:

```
9634.4
-825.0
```

COMPLETE

Your Response

Status

Your response has been submitted. You will receive your grade after all steps are complete and your response is fully assessed.

Your response

```
import requests
import json
import sys
import math

def due_amount(driver_id, joindate):
    #processing trips
    trips = requests.get("https://www.jsonkeeper.com/b/DM5F").json()
    joinMonth = int(joindate[:2])
    filtered = [f for f in trips if f["driver_id"] == driver_id]

    trips_after_second_month = [f for f in filtered if int(f["trip_date"][:2]) > joinMonth + 1]
    trips_after_first_month = [f for f in filtered if int(f["trip_date"][:2]) == joinMonth + 1]

    earning_for_20_percent_commision = sum(d['trip_details']['fare'] for d in trips_after_second_month)
    earning_for_10_percent_commision = sum(d['trip_details']['fare'] for d in trips_after_first_month)

    toBePayed = (earning_for_20_percent_commision*0.2) + (earning_for_10_percent_commision*0.1)

    #processing payments
    payments = requests.get("https://www.jsonkeeper.com/b/9QRZ").json()
    filteredPayments = [f for f in payments if f["driver_id"] == driver_id]
    payedAmount = sum(d['amount'] for d in filteredPayments)
    #round off to 1 decimal place
    return round(payedAmount - toBePayed, 1)

file_name = sys.argv[1]
with open(file_name) as f:
    testcases = f.readline()
    for case in range(int(testcases)):
        case1 = f.readline()
        [driver_id,date] = case1.split(',')
        #strip to remove any extra spaces from the string
        print(due_amount(int(driver_id),date.strip()))
```

Test Case Result Breakdown

Test Cases Result: 1 / 1

Test Input	Your Output	Expected Output
5 10017, 04/01/2022 10002, 03/05/2022 10011, 01/01/2022 10020, 02/20/2022	9634.4 1232.8 -825.0 6592.6 880.2	9634.4 1232.8 -825.0 6592.6 880.2

Previous | Next

Note: Please **SUBMIT** each question individually before ending the exam to receive score.

TIME REMAINING 0:02:40 End Exam

Previous | Tourists Pairing | ShareDrive | Car Manufacturing Design Problem | MBS The Line | Next

Tourists Pairing

Tourism Development Corporation of Bali provides transportation services to groups of tourists traveling to the same destination for sightseeing. They have a fleet of different cars available which have different seating capacities respectively. For a given group of tourists, they want to calculate the number of vehicles required to efficiently accommodate all tourists. Since it's a free-of-cost service and all the tourists are traveling to the same destination, the company wants the vehicles to be filled up to their maximum capacity and they want to use as few vehicles as they can to minimize the operational cost.

There are two arrays given as input;

- T[ ] which represents the tourist array that contains the number of tourists in each group
- S[ ] which represents the seating capacities array that contains the number of seats available in each vehicle

Write a function that, given two arrays T[ ] of tourists and S[ ] of seating capacities, consisting of integer elements, returns the minimum number of vehicles needed to accommodate all the passengers.
Given:

- Each element of arrays T and S is an integer within the range [1....100]
- T[K] <= S[K] for each K within the range [0..N-1]

Input

The input will be read from a file. The filename/path is passed to your program as the first command-line argument. The file will have two integer arrays, T[ ] and S[ ]. The first line of the file will contain a list of tourists and the second line will contain a list of seating capacities in each vehicle.

Output

The output will be an integer. (Minimum number of vehicles needed)

Examples:

Example 1. Given T = [4, 4, 2, 4] and S = [5, 5, 2, 5], the function should return 3.

Input
```
4, 4, 2, 4
5, 5, 2, 5
```

Output
```
3
```

Explanation:

5 tourists from group number 0 and 1 can travel in car number 0 with 5 seating capacity.
5 tourists from group number 1 and 2 can travel in car number 1 with 5 seating capacity.
The remaining 4 tourists can travel in car number 3 with 5 seating capacity.
Example 2. Given P = [2, 3, 4, 2] and S = [2, 5, 7, 2], the function should return 2.

Input
```
2, 3, 4, 2
2, 5, 7, 2
```

Output
```
2
```

Explanation:

5 tourists from group number 0 and 1 can travel in car number 1 with 5 seating capacity.
6 tourists from group number 2 and 3 can travel in car number 2 with 7 seating capacity.

COMPLETE

Your Response

Status

Your response has been submitted. You will receive your grade after all steps are complete and your response is fully assessed.

Your response

```
#include<iostream>
#include<sstream>
#include<vector>
#include<algorithm>
#include<numeric>
#include<fstream>

using namespace std;

void extractIntegersFromString(string &stringOfInts, vector<int> &integers)
{
    char seprator;
    int intValue;
    stringstream stream(stringOfInts);
    while(stream >> intValue) {
        integers.push_back(intValue);
        stream >> seprator;
    }
}

int main(int argc, char* argv[])
{
    ifstream inputFile(argv[1]);
    vector<int> tourists,carCapacities;
    string touristsInput, carCapacitiesInput;

    getline(inputFile,touristsInput);
    getline(inputFile,carCapacitiesInput);

    extractIntegersFromString(touristsInput, tourists);
    extractIntegersFromString(carCapacitiesInput, carCapacities);

    sort(carCapacities.begin(), carCapacities.end(), greater<int>());
    int touristsToBeAllocated = accumulate(tourists.begin(),tourists.end(),0);

    int requiredVehicles = 0;
    int index = 0;
    while(touristsToBeAllocated > 0)
    {
        if(index > carCapacities.size())
        {
            cout<<-1<<endl;
            return -1;
        }
        requiredVehicles++;
        touristsToBeAllocated -= carCapacities[index++];
    }
    cout<<requiredVehicles<<endl;
}
```

Test Case Result Breakdown

Test Cases Result: 3 / 3

Test Input	Your Output	Expected Output
3,2,3,2 5,2,1,8	2	2
4, 4, 2, 4 5, 5, 2, 5	3	3
2, 3, 4, 2 2, 5, 7, 2	2	2

Previous | Next