**Challenge**  
  
In finance, it's common for accounts to have so-called "velocity limits". In this task, you'll create a web application that accepts or declines attempts to load funds into customers' accounts in real-time.

In this application, you should provide a web page including:

- An interface to upload the input file for processing.  
- A data table to display the results after processing. User should be able to sort data in each column.

Each attempt to load funds will come as a single-line JSON payload, structured as follows:  
  
{  
  "id": "1234",  
  "customer\_id": "1234",  
  "load\_amount": "$123.45",  
  "time": "2018-01-01T00:00:00Z"  
}

Each customer is subject to three limits:  
  
A maximum of $5,000 can be loaded per day  
A maximum of $20,000 can be loaded per week  
A maximum of 3 loads can be performed per day, regardless of amount  
As such, a user attempting to load $3,000 twice in one day would be declined on the second attempt, as would a user attempting to load $400 four times in a day.

For each load attempt, you should return a JSON response indicating whether the fund load was accepted based on the user's activity, with the structure:  
  
{ "id": "1234", "customer\_id": "1234", "accepted": true }

You can assume that the input arrives in ascending chronological order and that if a load ID is observed more than once for a particular user, all but the first instance can be ignored. Each day is considered to end at midnight UTC, and weeks start on Monday (i.e. one second after 23:59:59 on Sunday).  
  
Your program should process lines from input.txt and display all the responses in the format specified above to a data table. The expected output given our input data can be found in output.txt.