Fund Load Restrictions Processing Challenge

In the financial services industry, it's critical to manage the flow of funds into customer accounts while adhering to specific velocity limits and other controls. This challenge involves creating a program to adjudicate fund load attempts based on specified limits and restrictions.
If you encounter any challenges or issues which require you to make assumptions about the data or challenge please document those assumptions as comments in your code.

Velocity Limits

Limit Type	Description
Daily Limit	A customer can load a maximum of $5,000 per day.
Weekly Limit	A customer can load a maximum of $20,000 per week.
Daily Load Count	A customer can perform a maximum of 3 load attempts per day, regardless of the amount.

Extra Credit - Special Sanctions (optional)

Sanction	Description
Prime ID	Due to new government regulations, prime numbers are suspicious. Only one load where 'id' is a prime number is allowed per day (across all clients), with a maximum amount of $9,999.
Mondays	The regulator has determined that Mondays are not ideal and so any loads which occur on this day are counted as double their value.

Input

Fund load attempts will be provided in a single-line JSON format. The input data will be provided in a file named input.txt

Output

For each fund load attempt, your program should output a valid JSON response indicating whether the load was accepted or declined based on the velocity limits.

Submission Format

```
{"id":"15337","customer_id":"999","accepted":false}
{"id":"34781","customer_id":"343","accepted":true}
{"id":"26440","customer_id":"222","accepted":true}
{"id":"12394","customer_id":"133","accepted":false}
{"id":"15689","customer_id":"445","accepted":true}
{"id":"32551","customer_id":"264","accepted":true}
{"id":"27446","customer_id":"328","accepted":true}
```

Please submit your output as valid JSON in a file called output.txt along with your code as a zip file. Please only upload the files that are relevant to evaluate your code. Also, please do not upload your `.git` history or leak any of your secrets in `.env` files or other configuration files.

Submission expectations:

We would like to get a signal on how you would build this software in an enterprise environment. We would be looking for well designed, well architected and testable software. Pretend you are submitting this solution as a pull request to be reviewed by your teammates. The review will be done as a debrief interview with Taxdome engineers. Think about maintainability, extensibility and scalability. But don't overdesign and don't build a distributed or parallel system. We would like to see some automated tests. You don't need to worry about extensive error handling (like file io etc…).

You may use any tools, including LLMs but if you do, please tell us about it and share your prompts. We recommend spending 2 hours solving this challenge, but we don't judge you on the time it takes.

We hope this helps and you enjoy building this project. Don't hesitate to ask us any questions.