MAIN OBJECTIVE

Write a script in JavaScript or PHP that reads two json data files and prints results to **stdout**. Your solution should work with any files that follow the same format as the provided example files.

GUIDANCE

- Spend **up to 3 hours**. If you run out of time, stop and tell us what you'd do next.
- We won't penalize the use of AI, in fact we encourage its responsible use. Thus, as part of evaluating your solution we'd like to see what prompts you used. **If you used AI to complete this assessment, please include a file `prompts.txt` with your solution.**

FILES YOU GET

`event_details.json` - *an array of events, each with ids for reference and details for evaluation.*

Layout*:*

```
[
  {
    "id_a": 13022,
    "id_b": 1071,
    "date": "2022-05-16",
    "value": 25,
    "color": "RED"
  },
  …
]
```

`references.json` - *an array of references, each having a name and an association id referencing the other file's events.*

Layout*:*

```
[{"name":"K", "id_a":13022}, …]
```

WHAT TO BUILD

Task 1: Color frequencies by month

Using only `event_details.json`, print the number of appearances of each **`color`** value in events having a **`date`** within the month of:

- **June 2024**
- **March 2025**

  Dates are in the format `YYYY-MM-DD`.

Task 2: Work only with referenced events

Match rule: *Each `references.json` entry references* ***event_details*** *by* ***`id_a`*** *or* ***`id_b`****. Assume `event_details.json` has* ***at most one*** *event for any given `id_a` and also* ***at most one*** *for `id_b`.*

Using the set of **event_details** which are connected to an entry in `references.json` by the above match rule, retrieve:

1a. **The sum of the events' `value` property**
Print the total.

1b. **Earliest date & max value → names**
Ties: If there are multiple, print the name which appears earlier in the alphabet.

- Find the **earliest `date`** property among these events. Print the **`name`** from the `references.json` entry that references it.
- Find the **minimum `value`** property among these events. Print the **`name`** from the `references.json` entry that references it.

1c. **High value threshold filter**
Filter the events down to only those with **`value`** `> 25`. For each remaining event print the **`name`** from the `references.json` entry that references it.

OUTPUT FORMAT

Your solution should be given in a directory containing two or more files:

- `solution.php` or `solution.js`
- `DESIGN.md`
- If you've used AI: `prompts.txt`
- Any other files necessary

Running your solution file should print **one JSON object** in the following form exactly:

```
{
  "task1": {
    "color_freq_2024_06": { "RED": 0, "BLUE": 0, "GREEN": 0 },
    "color_freq_2025_03": { "RED": 0, "BLUE": 0, "GREEN": 0 }
  },
  "task2": {
    "sum_value": 0,
    "earliest_date_name": "",
    "min_value_name": "",
    "high_value_names": ["..."]
  }
}
```

EVALUATION

We are looking for the following things in your solution:

1. Readability over brevity.
2. Some attention to reusability and centralization of logic.
3. A short `DESIGN.md` file with a few sentences that explains your approach, any trade-offs, and what you'd do next.

HOW TO SUBMIT

Share your solution with zach.sisti@realtyads.com and james@realtyads.com by doing one of the following:

1. Share a GitHub repository with appropriate access to checkout
2. Upload your solution to a Google Drive folder and share access