ET Phone Home - Decoder

Instructions

ET is trying to call home, but he forgot the combination for the intergalactic planet protocol! All he knows are the possible symbols and the number of symbols used for his home planet. The intergalactic routing service also scrambles the combination every 10 Earth minutes to prevent unauthorised use by space pirates.

ET does not have access to the galactic phonebook, so you have to help ET find the correct combination of symbols to call home!

Expose a `POST` endpoint `/decoder` for us to verify!

Input

You will be given the following with every attempt:

1. Possible values
2. No. of slots
3. History of previous attempts and results (a single integer value representing the below)
   i. Number of **right** symbols in the **wrong** position
   ii. Number of **right** symbols in the **right** position

```
{
  "possible_values": ["a", "b", "c", "d", "e", "f"],
  "num_slots": 4,
  "history":
   [
      {
         "output_received": ["a", "b", "c", "d"],
         "result": 22 // right symbol wrong position: 2, right symbol right position 2
      }
   ]
}
```

Output Expected

```
{
  "answer": ["a", "b", "d", "c"]
```

```
  }
```

Rules

1. Symbols can be repeated
2. The combination will change every 10 minutes
3. The highest score achieved is maintained so try as many times as you want
4. More points are given for guessing within fewer attempts

Examples

Case: All correct

If the correct combination is `["a", "b", "c", "d"]`

```
{
  "output_received": ["a", "b", "c", "d"],
  "result": 4 // right symbol wrong position: 0, right symbol right position: 4
}
```

Case: Repeated symbols

If the correct combination is `["a", "a", "a", "b"]`

```
{
  "output_received": ["a", "b", "a", "d"],
  "result": 12 // right symbol wrong position: 1, right symbol right position: 2
}
```

Case: Symbols in the wrong position will not be counted twice

If the correct combination is `["z", "c", "a", "c"]`

```
{
  "output_received": ["z", "a", "d", "a"],
  "result": 11 // right symbol wrong position: 1, right symbol right position: 1
}
```

Case: When 0 right symbol wrong position and 0 right symbol right position

If the correct combination is `["a", "b", "c", "d"]`

```
{
  "output_received": ["e", "e", "e", "e"],
  "result": 0 // right symbol wrong position: 0, right symbol right position: 0
}
```

Case: When 1 right symbol wrong position and 0 right symbol right position

If the correct combination is `["a", "b", "c", "d"]`

```
{
  "output_received": ["e", "e", "e", "a"],
  "result": 10 // right symbol wrong position: 1, right symbol right position: 0
}
```

Case: When 0 right symbol wrong position and 1 right symbol right position

If the correct combination is `["a", "b", "c", "d"]`

```
{
  "output_received": ["a", "e", "e", "e"],
  "result": 1 // right symbol wrong position: 0, right symbol right position: 1
}
```