

Problem Statement: Write a program that accepts a map of all possible outcomes of an event along with their probabilities and every occurrence of the event would generate outcomes based on the given probabilities. This could be seen as a generalization of events like rolling of a dice (could be biased) or flipping of a coin (could be biased).

Examples

1. Rolling of a six-faced biased dice
 - a. Input: [{1, 10}, {2, 30}, {3, 15}, {4, 15}, {5, 30}, {6, 0}]
2. Flipping of a coin
 - a. Input [{"Head": 35}, {"Tail": 65}]

Rules

1. Input: Probabilities given are as integers and percentages.
2. Each occurrence of the event should only generate one of the outcomes given in input
3. The outcome of each occurrence is independent of that of others.
4. On observing a large number (say 1000) of occurrences, the probability distribution should roughly follow the given biasness.

Example

- Input: [{1: 35}, {2: 65}] ## 1=Head, 2=Tail
- Output:
 - On triggering the event 1000 times, Head appeared 332 times and Tail 668 times which is roughly inline with the biasness given.
 - This is just one of the possibilities.

Solution

```
function generateOccurrences(outcomes, numOccurrences) {
  const occurrenceCount = {};

  for (const outcome in outcomes) {
    occurrenceCount[outcome] = 0;
  }

  for (let i = 0; i < numOccurrences; i++) {
    const randNum = Math.random() * 100;
    let cumulativeProb = 0;

    for (const outcome in outcomes) {
      cumulativeProb += outcomes[outcome];
      if (randNum <= cumulativeProb) {
        occurrenceCount[outcome]++;
        break;
      }
    }
  }
}
```

```

    }

    return occurrenceCount;
}

// Example usage
const outcomes = { 1: 10, 2: 30, 3: 15, 4: 15, 5: 30, 6: 0 };
const numOccurrences = 1000;

const occurrenceCount = generateOccurrences(outcomes,
numOccurrences);
for (const outcome in occurrenceCount) {
    console.log(`${outcome} appeared ${occurrenceCount[outcome]}
times`);
}

```

Problem Statement

Write a program that accepts multiple mathematical expressions and evaluates each of them using any public Web API available. The program should display the result of each expression on the console.

Rules

1. No expressions should be evaluated in the code. All evaluations should be using the Web API.
2. You can assume different expressions that are compatible with the API you choose.
 - a. Example: Some API might use ^ operator for power some might use pow()

Example

- Input (every line is an expression, evaluate when “end” is provided as an expression)
 - 2 * 4 * 4
 - 5 / (7 - 5)
 - sqrt(5^2 - 4^2)
 - sqrt(-3^2 - 4^2)
 - end
- Output
 - 2 * 4 * 4 => 32
 - 5 / (7 - 5) => 2.5
 - sqrt(5^2 - 4^2) => 3
 - sqrt(-3^2 - 4^2) = 5i

Solution

```

async function evaluateExpressions(expressions) {
    for (const expression of expressions) {

```

```

    try {
        const response = await fetch(
`https://api.mathjs.org/v4/?expr=${encodeURIComponent(expression)}`
        );
        const data = await response.json();
        if (data.error) {
            console.log(
                `Error evaluating expression "${expression}": Math
Error`
            );
        } else {
            console.log(`Result of expression "${expression}":
${data}`);
        }
    } catch (error) {
        console.log(`Error evaluating expression "${expression}":
Math error`);
    }
}

// Example usage
const expressions = ["2 + 2", "3 * 5", "sqrt(16)", "10 / 0"];

evaluateExpressions(expressions);

```