JavaScript Extras

Problem 1 Preço Certo

A contestant plays a spinning game where:

- There is a wheel with **21 slots**, each containing a value ranging from **0 to 100**, in steps of 5:
$$\{0, 5, 10, \ldots, 95, 100\}$$
- The contestant spins the wheel **up to two times**.
  - If the **first spin is 100**, the player **does not spin again**.
  - Otherwise, they must spin a **second time**.
- The contestant's final score is the **sum of the two spins**.
  - If the sum is **greater than 100**, the contestant is **disqualified** (score = 0).
  - If the sum is **less than or equal to 100**, it becomes their final score.

Given the two spin results for a player (`r1` and `r2`), calculate the **probability that they will win**, assuming:

- All valid spin combinations are equally likely (and independent).
- The player wins if their final score is the **highest among all valid combinations**.
- In case of a **tie for the highest score**, it is considered a **draw** (re-spin), and **not a win**, hence, if we get 100 total score, it shouldn't output 1 as in guaranteed victory (consider all other ways you can get 100).
- A final score **over 100** results in **disqualification** (score = 0).

Solution:

```
function chanceToWin(r1, r2) {
  const values = [];
  for (let i = 0; i <= 100; i += 5) values.push(i);
  const firstSpinValues = values.filter(v => v !== 100); // 20
     values (because if you get 100 in the first one, you're not
     playing the second round)
  const secondSpinValues = values; // 21 values (if you played the
     first round and got something under 100, any other value is a
     possibility)
  const allFinalScores = [];
  // 420 combinations (when the first spin is not= 100)
  for (let i = 0; i < firstSpinValues.length; i++) {
    for (let j = 0; j < secondSpinValues.length; j++) {
      const first = firstSpinValues[i];
      const second = secondSpinValues[j];
      const sum = first + second;
      if (sum > 100) {
        allFinalScores.push(0); // disqualified
      } else {
        allFinalScores.push(sum);
      }
    }
  }
  // +1 combination: first spin = 100 -> automatic 100 (there are
     421 combinations in total)
  allFinalScores.push(100);
  const playerScore = (r1 === 100) ? 100 : (r1 + r2 > 100 ? 0 : r1 +
      r2);
  const wins = allFinalScores.filter(score => score < playerScore).
     length;
  return wins / allFinalScores.length;
}
```