Probability Algorithm

Objective:

Calculate the probability of certain card/dice events.

Complexity level:

* Easy

Business Scenario:

* John Doe is playing a simple dice game trying to gain the lower number when two dice are rolled.

Problem Statement:

1. Two dice are rolled, die 1 (for player) and die 2(“opponent”). Both dice are rolled simultaneously. Create an algorithm that randomly “rolls 2 dice,” displays the numbers, and tells the player if he/she has lost or won.
2. Separately, calculate the probability of player 1 winning (using probability trees) not an algorithm, but demonstrates knowledge of simple probability.

Expectation Outcome:

This will introduce the concept of trees with an introduction to probability and random events / (random number function).

Tools:

* Open office for designing the flow chart.

Reference URL:

1. [Diaconis, Persi](https://en.wikipedia.org/wiki/Persi_Diaconis); and Keller, Joseph B.; ["Fair Dice"](http://www-stat.stanford.edu/%7Ecgates/PERSI/papers/fairdice.pdf), The American Mathematical Monthly, 96(4):337–339, 1989 (Discussion of dice that are fair "by symmetry" and "by continuity").
2. Iverson, G. R.; Longcour, W. H.; et al.; Bias and Runs in Dice Throwing and Recording: A Few Million Throws, Psychometrika, vol. 36, no. 1, March 1971.
3. [Knizia, Reiner](https://en.wikipedia.org/wiki/Reiner_Knizia), Dice Games Properly Explained, Elliot Right Way Books, 1999, [ISBN 0-7160-2112-9](https://en.wikipedia.org/wiki/Special:BookSources/0716021129) .