

---

## Dice-25

*Dice-25* is a game in which each player is given a 5x5 grid, and one die is thrown twenty five times. After each throw, each players must place the die outcome in an empty cell of his or her grid. At the end of the game, the grids are scored according to the dice values contained in the rows and columns, and the player with the highest score wins.

How the rows and columns are scored is arbitrary, but known to all players at the beginning of the game. For example, players might decide to score rows as columns as follows:

Contains	Example	Score
Sequence	1, 2, 3, 4, 5	80
Five	2, 2, 2, 2, 2	70
Four	3, 3, 3, 3, 1	50
Three Plus Two	4, 4, 4, 5, 5	30
Three	4, 4, 4, 5, 6	20
Two Plus Two	3, 3, 1, 6, 6	15
Two	4, 4, 1, 2, 3	5
Nil	1, 2, 3, 4, 6	0

### Part A

Write a function that takes in a row or a column of dice outcomes, and determines a score using the table on the first page of these notes.

### Part B

Write a function that takes in a filled board of dice outcomes and uses the function from part A to determine the total score for the board.

## Part C

It's actually not too difficult to use background simulations to get the computer to play a pretty good game of *Dice-25*. For example, to decide where to place a die outcome, a computer could do something like:

Given:

Die throw outcome D  
Board B

For each empty square E of board B:

Repeat many times:

Create a copy of board B; let's call the copy C.

Place outcome D in square E of C.

Place a random die throw outcome in each of the remaining empty squares of C.

Score board C,

Calculate the mean score.

Return the square with the highest mean score.

- Write a function that takes in a die throw outcome and a board, and returns the square where it would like to place the die outcome. You might want to start off with a smaller number of simulations when there are many empty squares on the board, and increase the number of simulations as the board is filled up.

The example on the next page shows the output of a program in which the user throws a die and tells the computer what the die outcome is. After each throw, the computer decides where in its grid to place the die outcome. After 25 throws, the computer calculates its score.

- Write an implementation of this program. Your program must choose which cell to place the die outcome in within five seconds. (Keep your implementation secret because we will have a competition to see whose implementation plays the best game.)

## Dice-25 Computer Competition

The competition will work as follows:

- Mr Ambler will throw a die 25 times, calling out the outcome after each throw.
- All contestants will tell their implementation what the die outcome is, and the implementation will place the outcome on a grid.
- After 25 throws, the implementation with the best score wins.
- We will play for the best of three games.

What is the next die? 2

```

.---.---.---.---.
|   |   |   |   |
:---:---:---:---:
|   |   | 2 |   |
:---:---:---:---:
|   |   |   |   |
:---:---:---:---:
|   |   |   |   |
:---:---:---:---:
|   |   |   |   |
:---:---:---:---:

```

What is the next die? 2

```

.---.---.---.---.
|   |   |   |   |
:---:---:---:---:
|   |   | 2 |   |
:---:---:---:---:
|   |   | 2 |   |
:---:---:---:---:
|   |   |   |   |
:---:---:---:---:
|   |   |   |   |
:---:---:---:---:
|   |   |   |   |
:---:---:---:---:

```

What is the next die? 1

```

.---.---.---.---.
|   |   |   |   |
:---:---:---:---:
|   |   | 2 |   |
:---:---:---:---:
|   |   | 2 |   |
:---:---:---:---:
|   | 1 |   |   |
:---:---:---:---:
|   |   |   |   |
:---:---:---:---:
|   |   |   |   |
:---:---:---:---:

```

What is the next die? 3

```

.---.---.---.---.
| 3 |   |   |   |
:---:---:---:---:
|   |   | 2 |   |
:---:---:---:---:
|   |   | 2 |   |
:---:---:---:---:
|   | 1 |   |   |
:---:---:---:---:
|   |   |   |   |
:---:---:---:---:
|   |   |   |   |
:---:---:---:---:

```

What is the next die? 6

```

.---.---.---.---.
| 3 |   |   |   |
:---:---:---:---:
|   |   | 2 |   |
:---:---:---:---:
|   |   | 2 |   |
:---:---:---:---:
|   | 1 |   |   |
:---:---:---:---:
|   |   | 6 |   |
:---:---:---:---:
|   |   |   |   |
:---:---:---:---:

```

[LATER]

What is the next die? 3

```

.---.---.---.---.
| 3 |   | 2 | 5 | 4 |
:---:---:---:---:
| 5 | 2 | 2 | 2 | 4 |
:---:---:---:---:
|   | 5 | 2 | 3 | 4 |
:---:---:---:---:
| 5 | 1 | 1 | 3 | 1 |
:---:---:---:---:
| 6 | 6 | 6 | 6 | 4 |
:---:---:---:---:
|   |   |   |   |
:---:---:---:---:

```

What is the next die? 4

```

.---.---.---.---.
| 3 | 4 | 2 | 5 | 4 |
:---:---:---:---:
| 5 | 2 | 2 | 2 | 4 |
:---:---:---:---:
|   | 5 | 2 | 3 | 4 |
:---:---:---:---:
| 5 | 1 | 1 | 3 | 1 |
:---:---:---:---:
| 6 | 6 | 6 | 6 | 4 |
:---:---:---:---:
|   |   |   |   |
:---:---:---:---:

```

What is the next die? 1

```

.---.---.---.---.
| 3 | 4 | 2 | 5 | 4 |
:---:---:---:---:
| 5 | 2 | 2 | 2 | 4 |
:---:---:---:---:
| 1 | 5 | 2 | 3 | 4 |
:---:---:---:---:
| 5 | 1 | 1 | 3 | 1 |
:---:---:---:---:
| 6 | 6 | 6 | 6 | 4 |
:---:---:---:---:
|   |   |   |   |
:---:---:---:---:

```

Row	Result	Score
1	two	5
2	three	20
3	sequence	80
4	three	20
5	four	50

Column	Result	Score
1	two	5
2	nil	0
3	three	20
4	two	5
5	four	50

Final score: 255