

PROJECT DUE
WEDNESDAY
 Apr. 29th

Factor Betting

You are writing a game, a betting game.

This game will revolve around betting on the outcome of rolling two dice. But there is a twist.



Here is an example:

- The user chooses a number, any positive integer. This number has several factors, some of which are possible dice rolls.
- For example, the number 20 has the following factors: 1,2,4,5,10,20.
 - The only possible dice rolls however are: 2,4,5,10
 - Which make up 11 possible rolls when using two dice.
 - This makes for a probability of $\frac{11}{36} = 31\%$

	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

- Now, it's betting time.
- Once you have established a percentage for betting, the user makes a wager. They start with a wallet of \$100 to bet from.
- Betting works by having three tries to roll the dice. The dice are rolled and the sum is found. If that sum is one of the factors, the user wins!
 - No matter the outcome, the bet is spent.
 - If the user wins, divide their bet by their probability (found above), and add it to their wallet.
 - Suppose the example above had a bet of \$20, and they won. The total won would be $\frac{\$20}{0.31} = \65

Requirements

- This game should be in a menu that repeats until a total of 15 rolls of the dice are completed, or the money has run out. This means there can be at most 5 cycles of choosing a new number and rolling three times each.
- The program should be user friendly and straightforward to use.
- There should be a constantly updating display of the user's wallet.
- The user's wallet, betting and winning amounts should all be treated as integers.
- There are at least three functions that are required. And they must be used as defined:

```
void displayStats(char name[], int wallet);
float bettingPercentSuccess(int sum);
int rollDie();
```

See my example program to test logic.
 It's ugly, but it works to test the dice rolling logic.