Going to Boston

Carefully read the following specification and submit a solution that implements that specification.

This is a "bar betting game" arranged from a traditional "Going to Boston" version. You will play with a computer. At the beginning, you and computer have the initial allowance of $1,000 each, and the game will continue until one becomes broke...

On one turn, you and computer roll three dice and show the two numbers of them but one die number being hidden. Based on the odds you think of, you will bet $1 to $10. The outcome is called "Big Fish!" if the three numbers are identical such as 1-1-1 and 2-2-2.

If not Big Fish, set aside the highest die and re-roll the other two. Keep the highest die of the two and re-roll the last. The total of all three dice is scored, and the bet amount multiplied by the difference of yours and computer's will be added/subtracted to each allowance. If Big Fish, the bet amount will be multiplied by 50 and added/subtracted.

If both the human and computer have the same sum, you should ouput the message "Tied" and print out unchanged money holdings (allowance) on hand for you and the computer. And then, for you and the computer, both should have a new set of three die values.

The same thing applies if both the human and computer both get a "Big Fish". Regardless of their sums, it is a tie. You should print out "Tied" and for both print out the Big Fish message.

For this assignment and all others, you should **exactly duplicate all prompts and output messages.**
Do not make up your own message strings.
Though your implementation is where your creativity should shine.

The purpose of this assignment is to verify that you understand variables, random variables, conditional statements (if) and loops. In your implementation **do not** roll the dice (generate random values) before you need them. If someone got a "Big Fish" that player only needs three random numbers for a round. Also **do not** use arrays, vector or any other container class.

For determining the maximum die value, I want you to use multiple "if" statements. **Do not** use the max function. Before you use the "wheel", I want to make sure you can invent it.

You can write your own little functions. But I do not expect you to do so until I cover functions in lecture.

See the below example of a game session:

```
[Round 1] You rolled (2, 2, ?), machine rolled (4, 6, ?)...
How much are you going to bet: 10
You had (2, 2, 2) ... Big Fish!!!
```

```
Machine had (4, 6, 3)
-> Machine rolled (2 ,4) and then rolled (5) -> Machine scored 15.
You won :) You have $1500, machine has $500...

[Round 2] You rolled (2, 5, ?), machine rolled (5, 1, ?)...
How much are you going to bet: 12
 - your bet must between $1 and $10, type again: 4
You had (2, 5, 6)-> You rolled (3 ,5) and then rolled (3) -> You
scored 14.
Machine had (5, 1, 1)-> Machine rolled (4 ,4) and then rolled (3) ->
Machine scored 12.
You won :) You have $1508, machine has $492...

[Round 3] You rolled (2, 1, ?), machine rolled (6, 1, ?)...
How much are you going to bet: 2
You had (2, 1, 1)-> You rolled (5 ,6) and then rolled (4) -> You
scored 12.
Machine had (6, 1, 6)-> Machine rolled (6 ,5) and then rolled (3) ->
Machine scored 15.
You lost :( You have $1502, machine has $498...

[Round 4] You rolled (2, 1, ?), machine rolled (6, 3, ?)...
How much are you going to bet: 4
You had (2, 1, 4)-> You rolled (5 ,4) and then rolled (4) -> You
scored 13.
Machine had (6, 3, 3)-> Machine rolled (6 ,2) and then rolled (6) ->
Machine scored 18.
You lost :( You have $1482, machine has $518...

[Round 5] You rolled (3, 1, ?), machine rolled (1, 5, ?)...
How much are you going to bet: 5
You had (3, 1, 1)-> You rolled (4 ,6) and then rolled (3) -> You
scored 12.
Machine had (1, 5, 4)-> Machine rolled (5 ,4) and then rolled (4) ->
Machine scored 14.
You lost :( You have $1472, machine has $528...

[Round 6] You rolled (6, 6, ?), machine rolled (4, 3, ?)...
How much are you going to bet: 10
You had (6, 6, 3)-> You rolled (3 ,5) and then rolled (2) -> You
scored 13.
Machine had (4, 3, 6)-> Machine rolled (5 ,2) and then rolled (5) ->
Machine scored 16.
You lost :( You have $1442, machine has $558...

[Round 7] You rolled (1, 1, ?), machine rolled (6, 3, ?)...
How much are you going to bet: 10
You had (1, 1, 6)-> You rolled (2 ,1) and then rolled (1) -> You
scored 9.
Machine had (6, 3, 4)-> Machine rolled (4 ,5) and then rolled (2) ->
Machine scored 13.
You lost :( You have $1402, machine has $598...
```

After you have implemented a solution where you (a human) interactively enters the amount of a bet, use pre-processor directives to conditionally compile the program such that you are replaced by some code;

```
#ifndef AUTOBET
//prompt for bet
//read bet
#else
//logic for determining bet
#endif
```

That logic might involve:

- Placing a high bet if you are holding two matching numbers
- Place a low bet if the computer is showing two matching numbers.
- Otherwise, place a medium bet

What you do exactly is up to you.

Use meaningful variable names
Reasonable use of white space (blank lines) including indentation of the body of conditional and looping statements

Assume proper user input: when a number asked is for, the user enters an appropriate numeric value and not a character or string value. We will deal with such mismatched input later in the course when we cover istream.

For this assignment and all others, you will be graded according to a rubric. Also, significant deductions will be made for late submissions.

When you submit your file.cpp, have the code so that when it is first compiled, the user will be entering bets via the keyboard.