CSI 201
Functions - Putting things together - more practice

1. We've had a worksheet on creating prototypes and attaching those to function calls. We've had a worksheet on function definitions. Then we had a worksheet on just using pre-existing functions! Today, we're going to do it all (or at least see it all) with a bunch of examples and test work.

2. Here's where I ultimately want to get:

```
#include<iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int main() {
   srand(time(0));
   int number_of_test_trials = 1000000;
   int wins_by_option1 = 0;
   for(int i = 0; i < number_of_test_trials; ++i) {
      if(option1() > option2()) {
         wins_by_option1 += 1; //this means ++
      }
   }
   cout << "Option 1 wins ";
   cout << wins_by_option1 * 100.0 / number_of_test_trials;
   cout << "% of the time" << endl;
}
```

3. We're going to starty by writing a function that rolls a 6-sided die. We'll call this function d6.

4. What is the signature of this function?

5. What should the function definition look like?

6. Okay, let's use our d6 function, to write a function that rolls three six sided dice and returns their sum. Let's start with a prototype:

7. And now the definition:

8. Okay, now let's use our d6 function to write a function that rolls two six sided dice but return their sum plus three. What's the prototype look like?

9. Okay and now the definition:

10. In the original code, we called the sum of three option1. And we called the sum of two plus three option2. Let's make option 3 and option 4 and test those.

11. For option 3, let's roll 4 dice, but we take the sum of the biggest three. You can do this by finding the sum of all four and subtracting the minimum from that sum.

12. For option 4, let's roll 3 dice, but give the sum of all three plus 2.

13. Which one wins?!

14. What if we change option4 to be only plus 1?