CS 201R
Fall 2020
Program 1 – Beat That!

The first programming assignment is a simple program to let you get used to writing in C++. Beat That! is a simple game played with 2 dice. For this program, it'll be played between the user and the computer, and the user will always roll first.

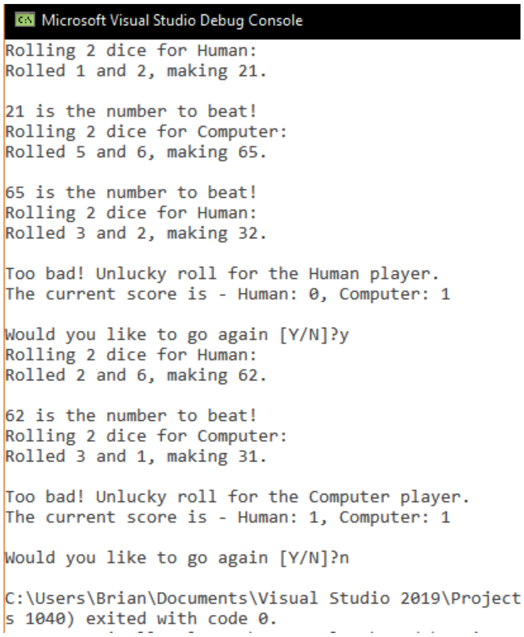
Each player rolls 2 dice, taking the results of the dice as digits, which should be arranged to make as large a number as possible. Rolling 2 1's would produce 11; rolling a 2 and a 3 could be 23 or 32, and so should be taken as 32, because it's larger; and so on. The largest possible number from 2 dice would be 66. After the player rolls, the computer and player take turns. Each player must roll dice forming a larger number than the last number their opponent rolled, otherwise the game is lost. Play continues until one player loses by rolling a number lower than their opponent's last roll; a game may just be one round, but it may be several. Your program should offer the user the choice whether to play again, and to record (and report) how many games have been won and lost by each side.

Your code should use a function to handle the dice rolling, reporting the dice, and finding the result. It must track whose turn it is, how many games have been won by each side.

Programming notes:

- C++ has a built-in random-number generator. By calling rand(), you can get a random positive integer. The range of these integers is limited, but still much bigger than 1-6. Use the modulo operator % to reduce it down to size.
- You will need to seed the random number generator with an initial value. The simplest way to do this is to use the system time by including the line:
  ```
  srand(0);
  ```
  early in main(). This will seed the generator with a particular value (in this case 0), and it will always return the same 'random' sequence. This is useful during testing. Changing the number you pass it will result in a different sequence, but the same seed will always produce the same sequence. When you are sure your program is working correctly, add the following at the top of your program with the other #include statements:
  ```
  #include <ctime>
  ```
  and change the call to srand() to:
  ```
  srand(time(NULL));
  ```
  This will seed the generator with the current time, and so produce a different sequence every time the program is run. (It will also generate a warning, because time() returns a value of type time_t, and srand() expects an unsigned integer. This warning can be ignored in this context.)
- Document preconditions, postconditions, return values, I/O, and any special instructions with each function header. Use appropriate variable names.

Submission:
Delete all contents of your Visual Studio "Debug" folder for the project, zip up the entire project folder, and upload the zip file to Canvas by the posted deadline. OR, if you're using version control such as GitHub, commit your changes, push them to the server, and submit the link on Canvas.



```
Rolling 2 dice for Human:
Rolled 1 and 2, making 21.

21 is the number to beat!
Rolling 2 dice for Computer:
Rolled 5 and 6, making 65.

65 is the number to beat!
Rolling 2 dice for Human:
Rolled 3 and 2, making 32.

Too bad! Unlucky roll for the Human player.
The current score is - Human: 0, Computer: 1

Would you like to go again [Y/N]?y
Rolling 2 dice for Human:
Rolled 2 and 6, making 62.

62 is the number to beat!
Rolling 2 dice for Computer:
Rolled 3 and 1, making 31.

Too bad! Unlucky roll for the Computer player.
The current score is - Human: 1, Computer: 1

Would you like to go again [Y/N]?n

C:\Users\Brian\Documents\Visual Studio 2019\Project
s 1040) exited with code 0.
```