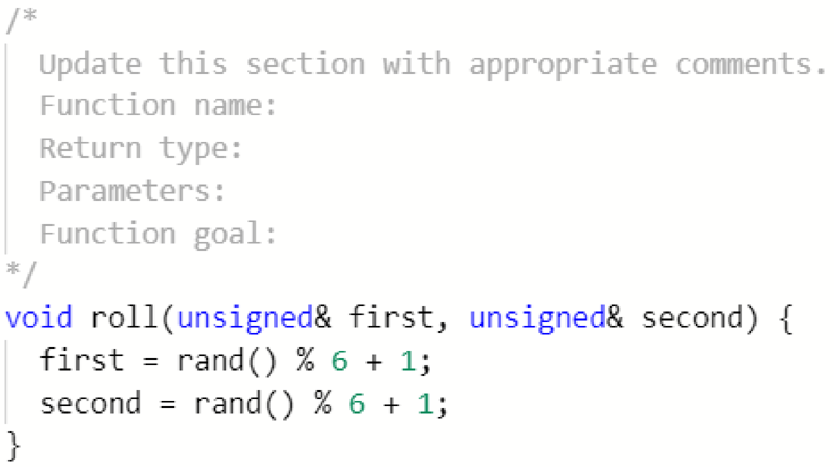
Part II Knock-out Game

As shown in this video, a player in a knock-out game rolls their dice multiple times and accumulates points according to the face values of the dice. The “knock-out” occurs when the face values of the dice add to 7. In that case, the points are reset to 0.

In this part of the assignment, you are asked to help one player roll their dice and keep their points.

You are asked to expand the ***week08_knockout*** project to first prompt the user for the number of times they would like to roll the dice. After that, your program shall set up a **counter-controlled loop** to repeat the following for each round:

- Call the following ***roll*** function to determine the dice values. Be sure to **update** the comment section with appropriate information. Do not change the roll function.

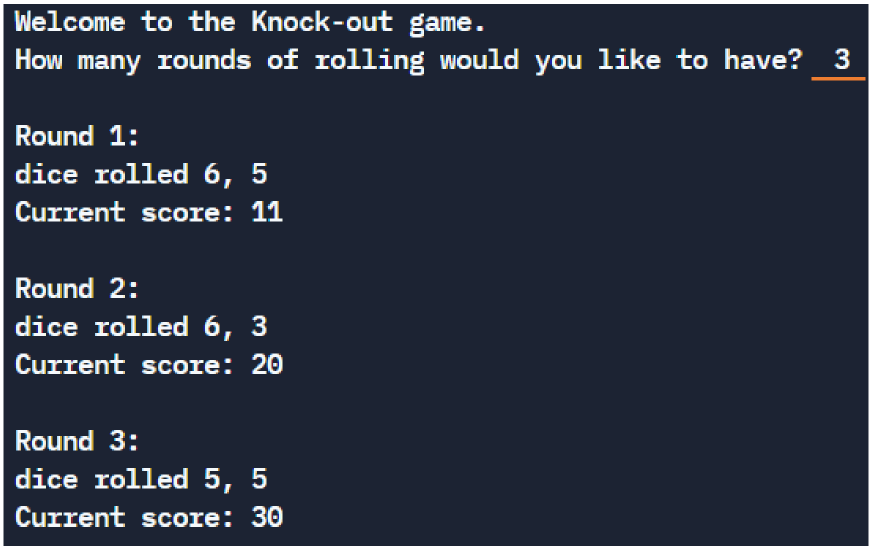


```
/*
  Update this section with appropriate comments.
  Function name:
  Return type:
  Parameters:
  Function goal:
*/
void roll(unsigned& first, unsigned& second) {
  first = rand() % 6 + 1;
  second = rand() % 6 + 1;
}
```

- Show the dice values as well as the updated score.

Here are some sample runs of the program.

The user was very lucky in the following run with no 7 being rolled.



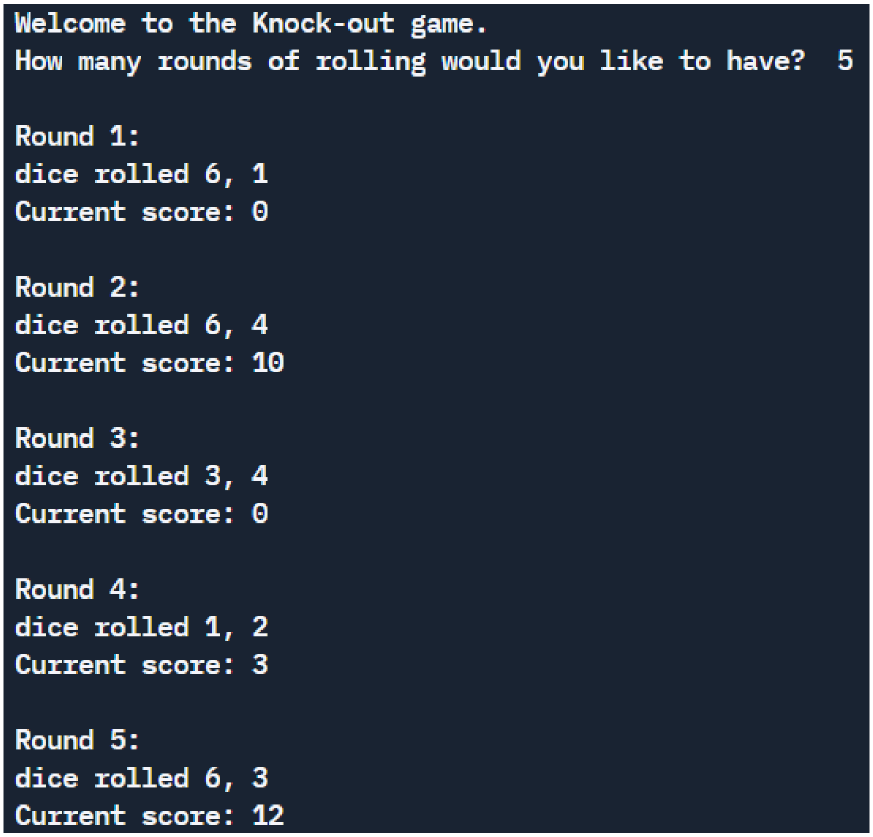
```
Welcome to the Knock-out game.
How many rounds of rolling would you like to have?  3

Round 1:
dice rolled 6, 5
Current score: 11

Round 2:
dice rolled 6, 3
Current score: 20

Round 3:
dice rolled 5, 5
Current score: 30
```

This one was not too bad. Although a 7 was rolled twice, they were not at the end of the game. Some points had been earned.



```
Welcome to the Knock-out game.
How many rounds of rolling would you like to have?  5

Round 1:
dice rolled 6, 1
Current score: 0

Round 2:
dice rolled 6, 4
Current score: 10

Round 3:
dice rolled 3, 4
Current score: 0

Round 4:
dice rolled 1, 2
Current score: 3

Round 5:
dice rolled 6, 3
Current score: 12
```