Assignment 11

26 November 2012

Due Monday 3 December at 1am

In this assignment, we will begin to implement a popular dice game called Yahtzee.™ In this game, players take turns rolling *five* six-sided dice. They can select which dice to re-roll, and then choose how to score the results.

Figure 1:

The score card looks like this (enlarge):

For this first part of the assignment, we will concentrate on writing functions that will detect certain categories of rolls, and score them.

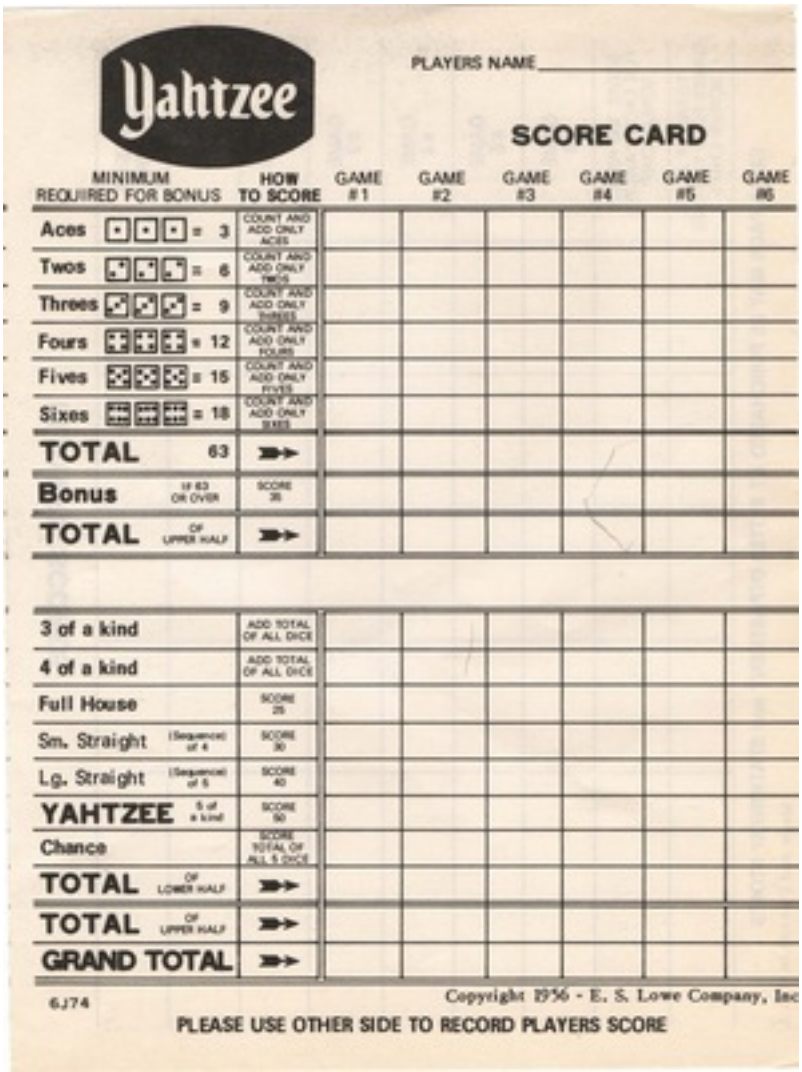
I have already given you these functions:

```
#include <iostream>
#include <stdlib.h>
using namespace std;

const int NUM_DICE = 5;

void rollDice(int dice[NUM_DICE])
{
    for(int i = 0; i < NUM_DICE; i++) {
        dice[i] = rand()%6+1;
    }
}

void showDice(int dice[NUM_DICE])
{
    cout << "Your dice: ";
    for(int i = 0; i < NUM_DICE; i++) {
        cout << dice[i] <<  " ";
    }
```



Yahtzee

PLAYERS NAME _____

SCORE CARD

MINIMUM REQUIRED FOR BONUS	HOW TO SCORE	GAME #1	GAME #2	GAME #3	GAME #4	GAME #5	GAME #6
Aces ⚀⚀⚀ = 3	COUNT AND ADD ONLY ACES						
Twos ⚁⚁⚁ = 6	COUNT AND ADD ONLY TWOS						
Threes ⚂⚂⚂ = 9	COUNT AND ADD ONLY THREES						
Fours ⚃⚃⚃ = 12	COUNT AND ADD ONLY FOURS						
Fives ⚄⚄⚄ = 15	COUNT AND ADD ONLY FIVES						
Sixes ⚅⚅⚅ = 18	COUNT AND ADD ONLY SIXES						
TOTAL 63	➤➤						
Bonus IF 63 OR OVER	SCORE 35						
TOTAL OF UPPER HALF	➤➤						
3 of a kind	ADD TOTAL OF ALL DICE						
4 of a kind	ADD TOTAL OF ALL DICE						
Full House	SCORE 25						
Sm. Straight (Sequence of 4)	SCORE 30						
Lg. Straight (Sequence of 5)	SCORE 40						
YAHTZEE 5 of a kind	SCORE 50						
Chance	SCORE TOTAL OF ALL 5 DICE						
TOTAL OF LOWER HALF	➤➤						
TOTAL OF UPPER HALF	➤➤						
GRAND TOTAL	➤➤						

6J74

PLEASE USE OTHER SIDE TO RECORD PLAYERS SCORE

Figure 2:

```
    cout << "\n";
}
```

which you can call from `main` like this:

```
    srand(time(NULL)); // Initialize random number generator
    int dice[NUM_DICE];
    rollDice(dice);
    showDice(dice);
```

Now you should implement and test the following:

```
// If there are three dice with the same value, return the sum
// of the values of all dice, else return zero.
int threeOfAKind(int dice[NUM_DICE]);

// If there are four dice with the same value, return the sum
// of the values of all dice, else return zero.
int fourOfAKind(int dice[NUM_DICE]);

// If there are three dice with the same value, return 50,
// else return zero.
int fiveOfAKind(int dice[NUM_DICE]);

// If there are three dice with the same value AND two dice with the
// same (but different) value, return 25, else return zero.
int fullHouse(int dice[NUM_DICE]);

// If there is a sequence of four dice (1234, 2345, or 3456), return
// 30, else return zero.
int smallStraight(int dice[NUM_DICE]);

// If there is a sequence of five dice (12345 or 23456), return 40,
// else return zero.
int largeStraight(int dice[NUM_DICE]);

// Return the sum of the dice that have the given value. So, if value
// is 4, then the sum of all dice whose value is four.
int sumOfValue(int dice[NUM_DICE], int value);
```

To test, you may want to do a bunch of random rounds and print the results:

```
    for(int i = 0; i < 20; i++) {
        rollDice(dice);
```

```
    showDice(dice);
    cout << "Three of a kind: " << threeOfAKind(dice) << "\n";
    cout << "Four of a kind:  " << fourOfAKind(dice) << "\n";
    // etc.
}
```

But you can also test specific cases like this:

```
int sampleFullHouse[NUM_DICE] = { 3, 4, 3, 3, 4 };
showDice(sampleFullHouse);
cout << "Full house: " << fullHouse(sampleFullHouse) << "\n";
    // should be 25
```