Assignment 2

Assignment Instructions:

- Complete the following programming exercises in C/C++.
- Questions 1-8 are worth 10 marks, Question 9 is worth 20 marks. (100 marks total)
- Aim to make your code as concise and logical as possible.
- For convenience, you can write all functions and class in a single file.
- Write functions to be self-contained units of code, i.e. the code inside your functions must **not** rely on variables you define outside the function (i.e. no *global variables*).
- Classes can be written as one block (i.e. there is no need to separate method declarations from their definitions).
- It's recommended that you test your code by calling your functions from your main() method, but no code written in main() will be graded.
- Do not use any library functions, unless the question specifies that you can.
- Submit your answers as a ***single text .TXT document***, *no zip or Word files etc.*, with **each answer clearly numbered and your code properly formatted/indented** (use Ctrl+A then Ctrl+I in Code::Blocks to properly format your code after you are finished writing it).

This assignment is due 6:00 pm Monday 24th February.

No late submissions accepted under any circumstances.

Q1. Write a function

```
unsigned int myStrLen(const char* str) { ... }
```

which operates similarly to the `strlen()` function in the standard C library, that is, it should return the number of characters in the C-style string `str` that is passed to it.

Q2. Write a function

```
double average(double data[], int size) { ... }
```

which takes an array of double values and calculates and returns their average. The "size" parameter specifies the length of the `data` array.

Q3. Write a function

```
bool isPrime(unsigned int num) { ... }
```

which returns true if `num` is a prime number and returns false otherwise. Note: 0 and 1 are not prime numbers.

Q4. By using **one for loop nested inside another for loop**, write a function `void twoDice() { ... }` that prints out all possible outcomes when of two six-sided dice are thrown together. When you call your function, the output should appear as follows:

```
(1,1) (1,2) (1,3) (1,4) (1,5) (1,6) (2,1) (2,2) (2,3) (2,4) (2,5) (2,6) (3,1) (3,2) (3,3) (3,4) (3,5) (3,6) (4,1) (4,2) (4,3) (4,4)
(4,5) (4,6) (5,1) (5,2) (5,3) (5,4) (5,5) (5,6) (6,1) (6,2) (6,3) (6,4) (6,5) (6,6)
```

Add some additional code to your function so that it also calculates and prints out the probability that the sum of two dice values will be 6 or greater.

Q5. Write a function `void simplifyFraction(int & n, int & d) { ... }` which takes a fraction in the form n/d and reduces it to its simplest form. For example, using the following test code in your main() function:

```
int numer = 234;
int denom = 832;
cout << numer << "\\" << denom << " = ";
simplifyFraction(numer, denom);
cout << numer << "\\" << denom << endl;
```

should print out the following:

```
2340\832 = 45\16
```

Q6. Using the binary operators >> and &, write a function

```
string toBinaryString(int a) { ... }
```

which returns a string of '0' and '1' characters that is the binary representation of the signed integer number a. For example, the following test code in your main() function:

```
cout << " 34 in binary is : " << toBinaryString(34) << endl;
cout << "-10 in binary is : " << toBinaryString(-10) << endl;
```

should print out

```
 34 in binary is 00000000000000000000000000100010
-10 in binary is 11111111111111111111111111110110
```

Q7. Write a **C++ class** called `Student`, which stores a student's first and last names (as separate private class members) and also stores the student's ID. Your class should also have the following public methods:

- a default constructor which sets the student's name to empty strings ("") and the ID to zero
- a constructor that allows a student object to be initialized with a first name, last name and ID
- a `print()` method, which prints out all student details

Q8. Write a separate function (not a class method), of the form

```
void printStudents(Student students[], int size) { ... }
```

which takes an array of `Student` objects (your C++ class from Q7.) and prints the details of each student in the array.

Q9. Write a C++ class called Circle which has x and y center coordinates and a radius as data members. The class should have the following features:

- A constructor which takes center coordinates x and y values and a radius value
- An area method which returns the area of the circle as a double value
- A display method which prints out the center, radius and area of the circle
- A method `bool intersects(const Circle & c)` which checks if this ciricle intersects with another circle (You can the maths library functions as needed)

Here's some test code for your main method you can use to to test your class:

```
Circle c1(0,0,1);
Circle c2(1,1,0.5);
c1.display();
c2.display();
cout << "Intersects? " << (c1.intersects(c2) ? " yes" : " no") << endl;
```

and the output it should produce:

```
center=(0,0) radius=1 area=3.41416
center=(1,1) radius=0.5 area=0.85354
Intersects?  yes
```

Submission status

Submission status	Submitted for grading
Grading status	Graded
Due date	Monday, 24 February 2020, 6:00 PM