ACMS 20210 Assignment 1

Due: 11:59 PM on Wednesday, February 15, 2017

Submit the C++ programs below using Sakai. If you are compiling from the command line on the CRC or another Linux system, I suggest using the command

```
g++ your_program_name.cpp -std=c++1y -o your_executable_name
```

This tells the compiler (called g++) to compile your .cpp source file into an executable file. To run this executable file from the command line, type

```
./your_executable_name
```

If you do not yet know how to use the CRC and do not have a compiler on your local machine, I suggest using the website *cpp.sh* to test your code (be sure to save a copy on your local machine, though!). You must submit your code in separate .cpp files.

1. Revise your program from the previous homework (Chapter 3, problem 11) to account for the paragraph beginning with "make some improvements." There are several ways of displaying the output to two decimal places. One way is to use doubles and the 'setprecision' i/o manipulator (look up the iomanip library and the setprecision manipulator). Another way involves using integer division and modular arithmetic. You may use whatever technique that you like.

2. Write a program that prompts the user to enter a positive integer N. The program should then compute the sum of the cubes of the integers from 1 to N inclusive *using a loop*. The program should also compute the sums of the cubes of the integers using the formula:

$$\sum_{k=1}^{n} k^3 = \left(\frac{(n)(n+1)}{2}\right)^2,$$

   and report this value to the user as well.

3. Write a program that prompts the user to enter a positive integer N. The program should then loop through all integer values from 1 to N inclusive. For each of the

integers, report whether the integer is divisible by 5, divisible by 7, divisible by both, or divisible by neither. For each integer, there should be at most one message displayed with cout. You can use the % operator to check divsibility.

4. Write a program to approximate π using the series

$$\pi = \sum_{k=1}^{\infty}(-1)^{k-1}\frac{4}{2k-1}$$

   Your program should prompt the user for an error tolerance and then compute enough terms in the series so that the error between your approximation and the true value of π is smaller than the tolerance. You can determine how many terms are required using the error bound in the alternating series test.

The following two problems are not due for submission, but you should complete them. I recommend using truth tables.

- Show that, for mathematical statements P, Q, we have the equivalence:

$$\sim((P\wedge \sim Q) \vee (Q\wedge \sim P)) \equiv (P \wedge Q) \vee ((\sim P) \wedge (\sim Q))$$

- Show that, for mathematical statements P, Q, R, we have the equivalence:

$$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$$