

ACMS 20210 Assignment 3

Due: 11:59 PM on Friday, September 28, 2018

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.

You may always assume that the user enters a reasonable value (e.g. if the problem states the user enters a positive integer, you can assume that they enter a positive integer).

Use the following naming scheme for submitting your solutions on Sakai: For exercise n on homework m , the name of your submission should be `hw_m_ex_n.cpp`. As an example, for homework 1 exercise 3, the name of your submission should be: `hw_1_ex_3.cpp`.

1. Write a program that prompts the user to enter a day, month, and year (as integers). The program then calls a function that returns a string representing the date in the format “Month Day, Year” (for example, if the user enters 10 for month, 29 for day, and 1929 for year, the function would return the string “October 29, 1929”). Print the nicely formatted date back to the user. You do not have to account for the possibility that the user enters an invalid date. (Hint: it is possible to convert the number for the month to the name of the month using a vector of strings. However, if we have not covered vectors by the time this assignment is due, conditionals will work fine.)

2. Write a program that loops through pairs of integers n and k , where n runs from 1 to 10 inclusive, and, for each n , k runs from 0 to n inclusive. For each of these pairs of values, call a function that computes $\binom{n}{k} = \frac{n!}{k!(n-k)!}$, and report the value (along with n and k) to the user.

3. Write a program that prompts the user for a positive integer N , and then iterates through the integers 1 to N inclusive, calling a function on each integer to test whether or not the integer is prime, reporting back to the user the result for each integer. Your prime testing function should accept an integer argument and return a Boolean value (true or false). Use trial division to test if the integer argument is prime. Make sure to incorporate a case for 1, which is not prime (you do not have to account for nonpositive arguments).

We will see a more efficient algorithm for finding all the primes from 1 to N in the future.

4. Write a program that prompts the user for a positive integer N . The program then computes $S(N) = \sum_{k=1}^N H(k)$ and reports this value to the user, where $H(k)$ is defined below.

For each number n , let $H(n)$ denote the number of times required to achieve an output of 1 by repeatedly applying the function $f(n)$ defined by:

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ 3n + 1 & \text{if } n \text{ is odd} \end{cases}$$

For example, for $n = 7$:

$$\begin{aligned} f(n) &= f(7) = 3(7) + 1 = 22 \\ f(22) &= 22/2 = 11 \\ f(11) &= 3(11) + 1 = 34 \\ f(34) &= 17 \\ f(17) &= 52 \\ f(52) &= 26 \\ f(26) &= 13 \\ f(13) &= 40 \\ f(40) &= 20 \\ f(20) &= 10 \\ f(10) &= 5 \\ f(5) &= 16 \\ f(16) &= 8 \\ f(8) &= 4 \\ f(4) &= 2 \\ f(2) &= 1 \end{aligned}$$

To achieve a value of 1, we had to apply f a total of 16 times, so $H(7) = 16$.

For $n = 1, 2, 3, 4, 5, 6, 7$, the values of H are 0, 1, 7, 2, 5, 8, 16 respectively. Additionally, $S(30) = 441$.

5. Let $f(n)$ be defined for integers n as below:

$$f(n) = \begin{cases} n & \text{if } n \leq 1 \\ 3 & \text{if } n = 2 \\ 2f(\frac{n}{2}) + 3 & \text{if } n \text{ is even and } n > 2 \\ f(\frac{n+1}{2}) + f(\frac{n-1}{2}) & \text{if } n \text{ is odd and } n > 1 \end{cases}$$

Write a program that prompts the user to enter a positive integer N and computes the following sum:

$$S(N) = \sum_{\substack{1 \leq n \leq N \\ f(n) \text{ is odd}}} f(n)$$

using a recursive function for f , reporting its value to the user.