MTH 3300
Spring 2017
E. Fink
March 18, 2017

Homework 6

Directions: Write the following 2 programs, and submit your source code to me via Blackboard, using the template files I have provided. You should send only the source code, in cpp format; do NOT send your .exe files. READ THE INSTRUCTIONS on how to submit your work in the Course Documents section of Blackboard.

This programming assignment is meant to give practice on loops and functions.

---

1) simpsons.cpp

If you've taken Calculus II, you've probably encountered Simpson's rule for approximating definite integrals:

$$\int_a^b f(x)dx \approx \frac{\Delta x}{3}\left(1f(x_0)+4f(x_1)+2f(x_2)+4f(x_3)+2f(x_4)+\ldots+2f(x_{N-2})+4f(x_{N-1})+1f(x_N)\right)$$

where

$$\Delta x = \frac{b-a}{N}$$

and

$$x_i = a + i\Delta x$$

Here, N is the number of pieces you break the interval $[a,b]$ into, which *must be even*; the more pieces, the better. Beware of the coefficients: they are 1 for the first term and the last term; all the rest alternate: 4,2,4,2,4,. . . , 4,2,4. (Again, if you want to know why this is a good approximation, we can talk, or consult a textbook).

So, for example: if I want to approximate $\int_1^2 x^4 dx$ with $N = 6$ steps, I have:

$$\Delta x = \frac{2-1}{6} = \frac{1}{6}$$

$$x_0 = 1,\ x_1 = 1+\frac{1}{6} = \frac{7}{6},\ x_2 = 1+\frac{2}{6} = \frac{8}{6},\ x_3 = 1+\frac{3}{6} = \frac{9}{6},\ x_4 = 1+\frac{4}{6} = \frac{10}{6},\ x_5 = 1+\frac{5}{6} = \frac{11}{6},\ x_4 = 1+\frac{6}{6} = 2$$

$$\int_1^2 x^4 dx \approx \frac{1/6}{3}\left(1(1)^4 + 4\left(\frac{7}{6}\right)^4 + 2\left(\frac{8}{6}\right)^4 + 4\left(\frac{9}{6}\right)^4 + 2\left(\frac{10}{6}\right)^4 + 4\left(\frac{11}{6}\right)^4 + 1(2)^4\right) \approx 6.2001028807$$

The exact answer is of course $\frac{31}{5} = 6.2$, so we're pretty close.

I want you to create a program that asks the user to enter in A, B and an even integer N, and uses Simpson's rule with N intervals to approximate the value of

$$\int_A^B \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx.$$

For instance, if $A = -1$, $B = 1$ and N is sufficiently large, this should be close to 0.68; if $A = -2$, $B = 2$ and N is large, this should be close to 0.95; if $A = -3$, $B = 3$ and N is large, this should be close to 0.997. (If those numbers sound vaguely familiar, that's because the given integral gives the area under the standard bell curve, between the z-values A and B.)

Hints: there is a special function in the math library for e^x, and π is equal to $4\arctan(1)$ – this will give more accuracy than typing in 3.14159. *Don't be intimidated by this problem* – it's just a big sum! But do be careful about the details, like initializing your sum, using the right number of terms, and providing the right coefficient for each term.

And finally, **make sure you understand Simpson's rule well enough to perform some calculations by hand**. It can be tedious, sure, but if you can't perform Simpson's rule by hand, you'll never be able to code it.

Specifications: your program must

- ask the user for values of A, B and N, where N is an even integer (you may assume the user complies).
- print an estimate for the value of $\int_A^B \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$ using Simpson's rule with N steps.

---

2) twins.cpp

Prime numbers (with the exception of 2) are always odd; so, aside from 2 and 3, the smallest possible difference between consecutive prime numbers is 2. For instance,

5 and 7 are primes, and 7 - 5 = 2;

11 and 13 are primes, and 13 - 11 = 2;

17 and 19 are primes, and 19 - 17 = 2;

etc. Two primes that differ by 2 are called *twin primes*. It has been known since antiquity that there are infinitely many prime numbers; however, it is still unproven that there are infinitely many twin prime pairs. (That is, it is unproven that the list that starts with (3,5), (5,7), (11,13), (17,19), (29,31), (41,43), ... *never ends.*) Interestingly, huge progress was made four years ago by an until-then-unfamous mathematician named Yiteng Zheng.

Your first mission is to create a function called `is_prime`. This function should take an integer as input, and return `true` if it has exactly two factors, and `false` otherwise. In my `twins.cpp` file, I have kindly put some test code, so that you can see if your function is working properly – none of the lines should say `ERROR!`.

Once your function is working, delete my code in `main()`. Your next mission is to create a program that asks the user to provide two positive integers P and Q. Your program should print all the twin prime pairs that lie between P and Q, inclusive. So, for example, if the user enters 11 and 43, the program should output `(11,13), (17,19), (29,31), (41,43)`. If the user enters 11 and 41, the program should just output `(11,13), (17,19), (29,31)`, since 43 isn't between 11 and 41.

The simplest way to do this is, roughly, go through the integers from P to Q; for each number, determine whether that number *and* that number plus 2 are both prime. Of course, it is up to you to get all the details right.

Specifications: your program must

- write a function called `is_prime`, which takes in an integer, and returns the value `true` if the integer is prime, and `false` if the integer is not prime. (It only needs to work for positive integers; note that 1 is technically not prime!)
- ask the user to enter in two positive integers, P and Q.
- print out all pairs of twin primes where both the primes are between P and Q inclusive.