**Name:**

**Advanced Programming in C++**

**Lab Exercise 4/15/2021**

For each of the following problems, submit your fully documented source code and sample output and fill in the answer as appropriate.

1. A binder weighs 1 lb. 5 oz. (One pound is 16 ounces). An empty shipping carton weighs 1 lb. 9 oz. and can hold up to 12 binders. The shipping costs include $1.44 for each full or partial carton plus $0.96 per pound or fraction of a pound plus a $3.00 service charge. Write a function using the following prototype to calculate the shipping cost as well as a main to test the function:

double shippingCost(int numberBinders);

1. The population of Mexico in 1990 was 89.2 million. Write a program that calculates and prints out the year in which the population of Mexico will reach 200 million, assuming a constant growth rate of 2.3% per year. Hint: Use a while loop.
2. The function rand() returns a random integer in the range from 0 to RAND\_MAX(32767). Using this function, write a program that generates a “fortune cookie” message consisting of a “fortune” and five “lucky numbers”. First, use a switch statement to randomly choose and display one of the following four “fortunes:”

You will have a long and happy life,

Never use break and continue in the same loop, or

A C++ programmer will live a long and prosperous life

Include cstdlib when using rand().

Then use rand() in a for loop to display the “lucky numbers” (five unique random integers from 1 to 99). Don’t forget to initialize your random number generator.

1. Write a void function that takes two integers as arguments, passed by reference, and replaces them with their sum and difference, respectively. Hint: be careful not to overwrite a value before you have used it.
2. Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Find the sum of all the even-valued terms in the sequence which do not exceed four million.

Answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. A Pythagorean triplet is a set of three natural numbers, *a b c*, for which,

*a*2 + *b*2 = *c*2

For example, 32 + 42 = 52. There exists exactly one Pythagorean triplet for which *a* + *b* + *c* = 1000. For that value of a, b, and c, find the product *a\*b\*c*.

Answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Zeller's congruence is an algorithm devised by Christian Zeller to calculate the day of the week for any Julian or Gregorian calendar date. It can be considered to be based on the conversion between Julian day and the calendar date.

For the Gregorian calendar, Zeller's congruence is

h = \left(q + \left\lfloor\frac{13(m+1)}{5}\right\rfloor + K + \left\lfloor\frac{K}{4}\right\rfloor + \left\lfloor\frac{J}{4}\right\rfloor - 2J\right) \mod 7,

where

* *h* is the day of the week (0 = Saturday, 1 = Sunday, 2 = Monday, ..., 6 = Friday)
* *q* is the day of the month
* *m* is the month (3 = March, 4 = April, 5 = May, ..., 14 = February)
* *K* the year of the century (year % 100).
* *J* is the [zero-based](https://en.wikipedia.org/wiki/Zero-based) century (actually year / 100) For example, the zero-based centuries for 1995 and 2000 are 19 and 20 respectively (to not be confused with the common ordinal century enumeration which indicates 20th for both cases).

**NOTE**: In this algorithm January and February are counted as months 13 and 14 of the previous year. E.g. if it is 2 February 2010, the algorithm counts the date as the second day of the fourteenth month of 2009 (14/2/2009 in MM/DD/YYYY format)

**NOTE**: In C++ the mod operator is %