CSE1301 Exercise Sheet 10
Numerical Computing and Lists: Search and Complexity

Exercise 1

(a) Represent the numbers 92, 40, 64, 3 and 17 in binary.

(b) Represent the ASCII characters 'a', 'A', and 'z' in binary.

Exercise 2

Represent the number –92 in

a) 8-bit signed magnitude
b) 8-bit 2's complement
c) 8-bit excess-*k*.

Exercise 3

Calculate, using binary arithmetic with 8-bit 2's complement representation:

a) 33+92
b) 33-92

What happens if you try to calculate 92+92?

Exercise 4

Suppose floating point numbers are represented in base 10, with 3-digit precision, and exponent in the range –16 <= exponent <= 15. Assume that numbers of greater precision are *rounded* to fit. Consider the real numbers:

$$x = 2.01513 \times 10^{12}$$
$$y = 4.92304 \times 10^{14}$$

Let x^* and y^* be the floating-point representations of x and y, respectively.

a) Give the mantissa and exponent of the floating point representation of $x^* + y^*$
b) Give a non-zero real number z such that $x^* + z^* = x^*$, where z^* is the floating-point representation of z in the form described above.

Exercise 5

Give the IEEE standard floating point representation of 1.11101101×2^{-5}.

Exercise 6

Give the 23-bit mantissa of the IEEE standard floating point representation of the decimal number 176. What is the sign bit of its IEEE standard floating point representation?

Exercise 7

a) Set up a list of numbers and construct a set of test data to test the binary search algorithm. Work through each example, making sure you understand how binary search works.

b) Modify the binary search function so that it returns the position of a sought number (rather than true) if the item is found, and it returns –1 if the item is not found.

c) Modify the binary search function so that it returns the position of a sought name (character string).

d) Modify the binary search function so that it finds a student by his/her ID number in an array of structs such as that defined in a struct of type ClassType (see Lecture 30).

e) What is the best case performance of binary search? What is the worst case performance?

Exercise 8 (Complexity)

A prime number is a natural number greater than 1 that can be divided without remainder only by itself and by 1. Write a simple algorithm to test whether a given number is a prime number. Code your algorithm in a C program. What is the runtime complexity of your algorithm?

The "Sieve of Eratosthenes" is one of the oldest methods for computing all prime numbers in a given range. Find out on the web about this algorithm and code it in a C program. What is the runtime complexity of your program? Could you improve the time complexity of your prime number test using the Sieve algorithm in it? Why or why not?

Exercise 9 (Additional)

Write a simple recursive algorithm to print out all prime factors of a given number.

Exercise 10 (Additional)

Which of the following standard laws of real number arithmetic may break down because of the limitations of the numerical representations of (a) `ints`, (b) `floats`? For the ones that may break down, give examples.

Commutative Law	$x + y = y + x$
Associative Law	$(x + y) + z = x + (y + z)$
Additive identity	$x + 0 = x$
Uniqueness of identity	$x + i = x$ implies $i = 0$
Additive inverse	for all x, there is a unique y such that $x + y = 0$
Distributive law	a*(x+y) = a*x+a*y

....