1806ICT Programming Fundamentals

Functions

1. Write a program to find the greatest common divisor (gcd) of two positive integers. The greatest common divisor of two integers is the largest integer value that evenly divides the two integers. Define a function that takes in the two integers and returns the gcd to the main program.

   Hint: Create a function gcd which takes 2 integer parameters a and b and returns the greatest common divisor of these 2 integers. In the function gcd have C code that does the following:

   - Let m equal the smallest of a and b
   - Loop with i starting at m and reducing down to 1. Stop when i evenly divides both a and b as i will be the greatest common divisor of a and b.

   Sample Run:

Input	Output
15 10	5
48 24	24

2. Write a coin tossing program. The program should use a function flipCoin( ) that returns 0 for tails and 1 for heads. Use a loop to flip the coin 1000 times, and count the number of times each side of the coin appears.

   Hint: Create a function flipCoin which randomly returns 0 or 1. Use the C rand() function to generate the random 0 and 1. Note that rand() randomly returns an integer in the range 1 to Maximum Integer so you will need to modify this to be a random 0 or 1.

3. An integer is said to be a perfect number if its factors, including 1 (but not the number itself), sums to the number. For example, 6 is a perfect number because 6 = 1 + 2 + 3. Write a function that prints out all perfect numbers between 1 and 1000.

4. Write a program that will convert a decimal number to its binary equivalent. Use functions in your program.

   Sample Run:

Input	Output
15	1111
27	11011

5. Write a function that alphabetically orders the stored values of three characters. Suppose for example that c1, c2, and c3 are character variables having the values ‘C’, ‘B’, and ‘D’, respectively. Then the function call

```
order_chars(&c1, &c2, &c3)
```

   should cause the stored values of c1, c2, and c3 to be ‘B’, ‘C’, and ‘D’, respectively.
   Write a program that tests your function.

6. Write a function that takes in a floating point number, and separates it into three parts: a sign (+ if it’s greater than zero, - if it’s less than zero, or blank if it’s exactly zero), a whole number magnitude, and a fractional part. Your function definition should look something like this:

```
void separate(double num, char *sign, int *wholePart,
              double *fracPart)
{
        …
    }
```

Write a program to test your function.

Sample Run:

Input	Output
39.212	Sign = + Whole number magnitude = 39 Fractional part = 0.212
-39.212	Sign = - Whole number magnitude = 39 Fractional part = 0.212
0.0	Sign = Whole number magnitude = 0 Fractional part = 0.0

7. Write a program that uses functions to evaluate how many times faster an average successful search will be in a sorted array of one million elements if it is done by binary search versus sequential search.