Lab 6

COMP9021, Session 1, 2014

The aim of this lab is to practice implementing functions, and in particular, functions that make use of the recursion technique.

1 A menu of operations

Write a program that displays a menu to add, subtract, multiply, or divide two floating point numbers, and that performs the operation. Users should first input `a`, `s`, `m`, `d` or `q`, and then two integer or floating point constants. They should be prompted again in case they fail to do so. If they choose division then they should be reminded that the second number they have to input should be nonnull. Use `%g` as the format specifier for the output. Here is a typical interaction.

```
$ a.out

Enter operation of choice:
a: addition          s: subtraction
m: multiplication    d: division
q: quit
a
Enter first argument to addition: 21
Enter second argument to addition: w21
Enter argument again: 12e2
21 + 1200 = 1221

Enter operation of choice:
a: addition          s: subtraction
m: multiplication    d: division
q: quit
r
Unknown operation, try again.

Enter operation of choice:
a: addition          s: subtraction
m: multiplication    d: division
q: quit
d
Enter first argument to division: -12e3
Enter second argument to division:
e
Enter argument again: 0
Argument cannot be null. Enter second argument to division again: 4
-12000 / 4 = -3000
```

```
Enter operation of choice:
a: addition           s: subtraction
m: multiplication     d: division
q: quit
q
$
```

2 ☞ Greatest common divisor

Recall Euclid's method to compute the *gcd* (*greatest common divisor*) of two nonnegative numbers a and b at least one of which is not equal to 0:

- if $b = 0$ then the gcd of a and b is a;
- if $a \geq b$ then the gcd of a and b is the gcd of b and the remainder of the division of a by b;
- if $a < b$ then the gcd of a and b is equal to the gcd of b and a.

For instance, the gcd of 60 and 45 is 15. Write a program that prompts the user for two positive numbers a and b of type `long long`, at least one of which is not zero and outputs the gcd of both numbers. Here is a possible interaction:

```
$ a.out

Input two integers: 18 1
The gcd of 18 and 1 is 1.
$ a.out

Input two integers: 34 32
The gcd of 34 and 32 is 2.
$ a.out

Input two integers: 32 35
The gcd of 32 and 35 is 1.
$ a.out

Input two integers: 32 36
The gcd of 32 and 36 is 4.
```

3 ☞ Obtaining a sum from a subsequence of digits

Write a program that prompts the user for two numbers: an `unsigned long long`, say `digits`, and an `int`, say `sum`, and outputs the number of ways of selecting digits from `digits` that sum up to `sum`. For instance, if `digits` is `12234` and `sum` is `5` then there are 4 solutions:

- one solution is obtained by selecting 1 and both occurrences of 2 $(1 + 2 + 2 = 5)$;
- one solution is obtained by selecting 1 and 4 $(1 + 4 = 5)$;
- one solution is obtained by selecting the first occurrence of 2 and 3 $(2 + 3 = 5)$;
- one solution is obtained by selecting the second occurrence of 2 and 3 $(2 + 3 = 5)$.

Here is a possible interaction:

```
$ a.out

Input an unsigned long long and an int: 12234 5
There are 4 solutions.
$ a.out

Input an unsigned long long and an int: 11111 5
There is a unique solution
$ a.out

Input an unsigned long long and an int: 11111 6
There is no solution.
$ a.out

Input an unsigned long long and an int: 1234321 5
There are 10 solutions.
```