Course: DD2325 - Exercise Set 3

Exercise 1: *Getting started with* `C`

Most people's first `C` program looks something like this:

```
#include <stdio.h>                  include information about the standard input/output library

main()                              define a function named main that receives no argument values
{
   printf("hello, class\n");        main calls library function printf to print to the screen
}
```

If this program is written in the file `hello.c` then it is compiled using
$ `gcc hello.c -o hello`
and run with the command
$ `./hello`

Note! Normally you are free to give functions whatever name you like, but `main` is special - your program begins executing at the beginning of `main`. This means that every program must have a `main` somewhere.

Adapt the above function so that it prints out your name and your birthday.

Exercise 2: *Loops*

Write a snippet of code that sums the numbers `1` to `n` using a (a) `for` loop, (b) `while` loop and (c) a closed form solution.

Exercise 3: *Loops*

Rewrite the following as `for`-loops:

```
int i = 1;
while(i <= 10){
   if (i < 5 && i != 2)
      printf("X");
   i++;
}
```

(a)

```
int i = 1;
while(i <= 10){
   printf("X");
   i = i + 3;
}
```

(b)

```
long m = 100;
do{
   printf("X");
   m = m + 100;
}while(m < 1000);
```

(c)

Exercise 4: *Loops*

Write a program that prompts the user for two postive integers `m` and `n`. Ask the user to make sure that `n > m`, but you can also get the program to check this condition. These numbers are read in by the program which then proceeds to print all the numbers divisible by `m` between `1` and `n`.

Exercise 5:

Write a program that tells what coins to give out for a user defined amount of change from 1 cent to 99 cents (assume you are in a Euro-zone country). For example, if the amount is 86 cents, the output would be something like the following:

```
86 cents can be returned as
1:50-cent, 1:20-cent, 1:10-cent, 1:5-cent, 1:1-cent
```

(**Hint** You can use arrays if you know about them.)

Exercise 6: *Loops*

What is the output of this loop? Identify the connection between the value of `n` and the variable of the variable l.

```
int n = 1024, i;
int l = 0;
for (i=1; i<n; i = i*2)
   l++
printf("n:  %d, l:  %d", n, l)
```

Exercise 7: *Loops*

What is the output of this loop? Comment on the code.

```
int n = 1024, i;
int l = 0;
for (i=0; i<n; i = i*2)
   l++
printf("n:  %d, l:  %d", n, l)
```

Exercise 8: *User defined functions*

Write a program that prompts the user for a postive integer `n` and then calls a function that sums the numbers from `1` to `n`.

Exercise 9: *Calculating e^x*

The value of e^x can be written as:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \approx \sum_{n=0}^{N} \frac{x^n}{n!}, \quad \text{for } N \text{ large enough}$$

Write a function that takes `x` and `N` as input and computes the sum.

Write a program that compares the accuracy of your function for calculating e^x and the predefined function `exp` in the library `math.h` for different values of `N`.

Suggest how you could find a decent value of `N` for a given value of `x`.

Exercise 10: *Fibonacci sequences*

In a colony of honeybees there is one special female called the **queen**. There are many **worker** bees who are female too but unlike the queen bee, they produce no eggs. There are some **drone** bees who are male and do no work. Males are produced by the queen's unfertilised eggs, so male bees only have a mother but no father. All the females are produced when the queen has mated with a male and so have two parents.

In summary female bees have 2 parents, a male and a female whereas male bees have just one parent, a female.

Consider the ancestors of a male drone bee. Let $A(n)$ represent the total number of ancestors, $M(n)$ the number of male ancestors and $F(n)$ the number of female ancestors n generations back. In the following table f represents a female relative and m a male one. So for the first four generations:

Generation Level (n)	ancestors	$M(n)$	$F(n)$	$A(n)$
1 (parents)	f	0	1	1
2 (grand-parents)	mf	1	1	2
3 (great-grand-parents)	fmf	1	2	3
4	mffmf	2	3	5
5	fmfmffmf	3	5	8

(a) What are the values of $M(6), F(6)$ and $A(6)$?

(b) What is the recurrence relation that generates the sequence of numbers represented by $F(n)$ and $M(n)$? Using these relations what is the recurrence relation generating the sequence of numbers defined by $A(n)$.

(c) Write a program in `C` that prompts the user for a value of n and then calculates $A(n)$.

Exercise 11: *Numerical integration - trapezoidal rule*

In mathematics, the trapezoidal rule is a way to approximately calculate the definite integral:

$$\int_a^b f(x)\,dx$$

The trapezoidal rule works by approximating the region under the graph of the function $f(x)$ by a trapezoid and calculating its area. It follows that:

$$\int_a^b f(x)\,dx \approx (b-a)\frac{1}{2}(f(a)+f(b))$$

To calculate this integral more accurately, split the interval of integration $[a,b]$ into n smaller subintervals, and apply the trapezoidal rule on each of them. One obtains the *composite trapezoidal rule*:

$$\int_a^b f(x)\,dx \approx \frac{(b-a)}{2n}\left(f(x_0)+2f(x_1)+2f(x_2)+\ldots+2f(x_{n-1})+f(x_n)\right)$$

where

$$x_k = a + k\frac{(b-a)}{n},\ \text{for } k = 0,1,\ldots,n$$

Write a program to evaluate the definite integral

$$\int_0^1 \sqrt{x^2+1}\,dx$$