

# ENEE 140 Lab 10

## Lab instructions

This handout includes instructions for the recitation sessions on Wednesday and Friday. **Follow these instructions** to review to review loops and file I/O, then **submit the homework** as indicated below. To prepare for the next lecture, complete the **reading assignment** and try to solve the **weekly challenge**.

### 1 Fun with loops

1. Read the following code:

```
#include <stdio.h>
#define line 111111111

int main(void){
    int i, j = 1;

    for (i=1; i<10; i++) {
        j = j * 10;
        printf("%9.*d.%d\n", i, line*i%j, line*i%j);
    }

    return 0;
}
```

2. Think about what will be the output and write it down
3. Type the code in CLion, compile it and execute it. Is the printout on the screen the same as you predicted in step b? If not, find the reason.
4. Try to implement the same functionality using a **while** loop.

### 2 File input/output

Go through `fileIO.c` carefully and pay special attention to the following:

1. `int main(int argc, char *argv[])`

This is the way you access arguments passed to the program on the command line.

2. **FILE** \*section, \*name, \*message;

This is the way to declare variables of **FILE** type. Pay attention to the \* in front of the variable names

3. name = fopen(argv[2], "r");

**fopen()**: command to open file; **argv[]**: array of strings that holds the command line arguments (**argv[0]** is the name of the executable program).

4. name = fopen("name.txt", "r");

This will open the file **name.txt**, located in the same directory as the executable file. If the file is at a different directory, the full path to the file needs to be provided here.

5. **exit(0)**;

A library function to terminate the execution. **stdlib.h** must be included in order to use this function.

6. **while** (fscanf(section, "%d", &number) != EOF)

Read the file for section number until the end of the file is reached (**EOF** = End Of File).

---

## Homework

**Due:** April 17 at 11:59 pm.

Create two programs by following the instructions below. Submit them using the following commands:

```
submit 2016 spring enee 140 AAAA 10 compute_stats.c
submit 2016 spring enee 140 AAAA 10 cmdline.c
submit 2016 spring enee 140 AAAA 10 cal.c
```

Note: you must replace **AAAA** with your own section number (0101, 0102, etc.)

### 1 Computing aggregates

Write a program, called `compute_stats.c`, which opens a data file, called `numbers.txt`, which contains a list of positive integers, computes their sum and product, finds the largest number, and writes the results to an output file called `statistics.txt`. Test your program by creating a `numbers.txt` file and by inserting several number combinations in this file.

For example, if data file has the following numbers

```
2 3 5 7 9 4 6 12
```

your program should write the following to the output file

```
sum = 48
product = 544320
largest = 12
```

### 2 Command line arguments

Write a program, called `cmdline.c`, that prints out different greetings based on the command line arguments.

- `./cmdline last_name`  
Hello, Mr. last\_name
- `./cmdline last_name first_name`  
My name is first\_name last\_name
- `./cmdline last_name test_score`  
Hi, Mr. last\_name, your score is test\_score

Some sample output:

```
Hello, Mr. Jones
My name is Joe Smith
Hi, Mr. Doe, your score is 87.5
```

### 3 Monthly calendar

The UNIX `cal` command prints out the calendar of the month/year that the user enters. Type in the following, one at a time, and observe the output:

```
cal 3 2014
```

```
cal 2014
```

```
cal 1 1
```

To learn more about this command, type in `man cal` for help.

Write a program named `cal.c` that prints out the following (which is the output when you type in `cal 4 2016` in UNIX). Note that you are not asked to implement the `cal` command. You can assume that you know April 1 is a Friday. You will need to use loops and the `%` operator.

```
    April 2016
Su Mo Tu We Th Fr Sa
           1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

## **Reading assignment**

K&R Chapters 6.2, 6.3, 6.7, 6.8, 8.1, 8.2, 8.3, 8.4.

## **Weekly challenge**

No challenge this week. Start working on Project 3, and think about to implement the various functions required. Next week, in class, we will discuss these implementation options.