ENEE 140 Lab 11

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:

```c
#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 the output will be 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 2? 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 (you can find `fileIO.c` in the class public directory). 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: Friday at 11:59 pm.

Create one program by following the instructions below.

1 File interleaving

Write a program, called `file_interleaving.c`, that receives three arguments from the command line:

```
file_interleaving file1.txt file2.txt file3.txt
```

Your program should open files `file1.txt` and `file2.txt` for reading and to create file `file3.txt` as follows:

- The first line of `file3.txt` is the first line of `file1.txt`
- The second line of `file3.txt` is the first line of `file2.txt`
- The third line of `file3.txt` is the second line of `file1.txt`
- The fourth line of `file3.txt` is the second line of `file2.txt`
- ...

When one input file reaches the EOF, the remaining lines in the other file should be copied to the output file and the program terminates. Your program should print appropriate error messages if fewer than 3 file names are provided on the command line or if the files cannot be opened.

Log into Elms, click on Gradescope in the course menu, then go to the relevant assignment to submit your work.

Reading assignment

Read K&R Chapter 5.11 (read about how to use qsort). Re-read chapters B11, 3.5.

Weekly challenge

Write a program that sorts an array of integers, so that `a[0] <= a[1] <= a[2] <= … <= a[n-1]`. In other words, implement the `selection_sort()` function so that the output of the program is as follows:

```
56 6 58 20 8 45 56 12 60
6 8 12 20 45 56 56 58 60
```

You can use the following template (also available in the GLUE class public directory, at `public/challenges/week12`):

```c
/*
 * selection_sort.c
 *
 *  Sort an array, by gradually building up the sorted array.
 *  At each iteration:
 *  - The beginning part of the array contains the lowest elements,
 *    in sorted order.
 *  - Find the minimum element in the unsorted part of the array.
 *  - Add it to the end of the sorted part.
 */

#include <stdio.h>


void
print_array(int n, int a[])
{
        int i;

        for (i=0; i<n; i++)
                printf("%d ", a[i]);

        printf("\n");
}


void
selection_sort(int n, int a[])
{
}


int
main()
{
        int a[100] = {56, 6, 58, 20, 8, 45, 56, 12, 60};

        print_array(9, a);
```

```
        selection_sort(9, a);

        print_array(9, a);

        return 0;
}
```

The weekly challenge will not be graded. However, if you manage to solve it, you may submit it for extra credit. The deadline for submitting your solution to the weekly challenge is **Monday at 11:59 pm**.

Submit it by going to the relevant assignment in Gradescope.