ENEE 140 Lab 7

Lab instructions

This handout includes instructions for the recitation sessions on Wednesday and Friday. **Follow these instructions** to understand common problems with `if` statements and to practice invoking the random number generator from the C standard library, 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 The dangling `else` problem

You are asked to write a program to read in two integers `x` and `y`, then check whether (i) both `x` and `y` are positive, or (ii) `x` is negative. Will the following code segment do the work? Check your answer by testing it in a complete program with the inputs provided below.

```
int x,y;
scanf("%d%d", &x, &y);
if (x > 0)
  if (y > 0)
    printf("Both positive.\n");
else
  printf("x negative, y ignored.\n");
```

1. x = 2, y = 1
2. x = 2, y = -1
3. x = -2, y = -1
4. x = -2, y = 1

What did you learn from this example?

2 `if-else`

What will be the output of the following code segment? Think about the answer and then check your answer by testing it in a complete program.

```
int a = 5;
if (a = 0) {
```

```
 printf("a=0 is true. \n");
}
else {
 printf("a=0 is not true.\n");
}
if (a==0) {
 printf("a==0 is true. \n");
}
else {
 printf("a==0 is not true.\n");
}
```

3 Random number generation

Write a C program that generates 6 random even numbers between 2 and 20 (including both 2 and 20) and prints them out.

Homework

Due: March 11 at 11:59 pm.

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

```
submit 2016 spring enee 140 AAAA 7 array_index.c
submit 2016 spring enee 140 AAAA 7 extract_letters.c
submit 2016 spring enee 140 AAAA 7 day_of_the_week.c
```

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

1 Array index

Write a program, called `array_index.c`, that asks the user to enter 10 positive integers, stores them in an array of size 10, then asks the user for another number `k`, which is between 1 and 10, and prints out the `k`th number that the user has entered. For example, when user enters

```
2 3 5 7 11 13 17 19 23 29
```

and then gives 3 for the value of `k`, you should output 5, the third number the user entered. The program should print an error message if `k` is not between 1 and 10.

2 Extract letters

Write a program, called `extract_letters.c`, that prompts the user to enter a string and then prints out the English alphabetic characters (both lower case and upper case) that the user has entered. Below are some sample input-output pairs (note the white spaces in the second example):

```
Input: Cat808her_+in_+e87                  output: Catherine
Input: 2012 Nov  06_ENEE140$$$exa !m       output: NovENEEexam
```

3 Day of the week

Assuming that Jan. 1 is Monday, write a program, called `day_of_the_week.c`, which prompts the user for a positive integer `k` and then determines and prints out the day of the week for the `k`th day of the year. For example, on `k` = 16, your output should look like:

```
The 16th day of the year is Tuesday.
```

Hint. Think about the relation between k%7 and the day of the week. Also, to keep it simple, when k = 1, 2, or 3 it is ok to print out 1th, 2th, or 3th instead of 1st, 2nd, 3rd.

Reading assignment

K&R Chapters 4.3, 4.4, 4.5, 4.6, 4.8, 4.9, 4.11

Weekly challenge

Write a program to remove trailing blanks and tabs from each line of input, and delete entirely blank lines.

You can use the following template (also available in the GRACE class public directory, at `public/challenges/week07`):

```
/*
 * trim_strings.c
 *
 *  Remove trailing blanks and tabs from each line of input, and
 *  delete entirely blank lines.
 *  K&R Exercise 1.18
 */


#include <stdio.h>
#include <string.h>
#include <ctype.h>


// Maximum input line size
#define MAXLINE 256


// Function that trims the whitespace at the end of string s
// and returns the length of the new string.
int trim_string (char s[]);


int
main()
{
        // Read input line-by-line
                // Trim whitespace and print line if any characters left

        return 0;
}


int
trim_string (char s[])
{
        // Implementation of the trim_string function
}
```

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. To be eligible for extra credit, you must implement correctly **all but two** of the weekly challenges. You can submit your program from a GRACE machine using the following command (replace `AAAA` with your section number):

```
submit 2016 spring enee 140 AAAA 1007 trim_strings.c
```