ENEE 140 Lab 8

Lab instructions

This handout includes instructions for the recitation sessions on Wednesday and Friday. **Follow these instructions** to review strings and arrays, 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 Array review

Go over `array.c` and answer the following questions (you can find `array.c` in the class public directory):

Question 1: Why define 10 as a constant variable SIZE instead of just using the value 10?

Question 2: Why initialize sum with 0 rather than 1 or -1?

The following 3 lines of code read in 10 integers one by one.

```
for (i=0; i<SIZE; i++) {
    printf("Enter integer number %d: ", i);
    scanf("%d", &a[i]);
}
```

Question 3: How is the sum of the 10 integers calculated?

Question 4: Add the following line after you print out sum, re-compile the code, run it. Observe the output and try to guess/reason why this happens.

```
printf("overflow: a[SIZE] = %4d\n", a[SIZE]);
```

2 `printf` with string arguments

The `%n.ks` format option in a `printf()` call will reserve n spaces in total but print out only the first k characters from the string. What will happen if the string has less than k characters? What would happen if k>n? Write a simple program to verify your answer.

Homework

Due: Friday at 11:59 pm.

Create one program by following the instructions below and

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

1 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
```

Reading assignment

K&R Chapters 2.11, 2.12, 3.4, 3.5, 3.6, 3.7, 3.8, 5.10, 6.2, 6.3, 6.7.

Weekly challenge

Write a program that reads a string and then checks if the string meets the quality rules for UMD Directory passwords.

You can use the following template (also available in the GRACE class public directory, at `public/challenges/week08`):

```
/*
 * check_password_rules.c
 */

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

#define MAX_PASSWORD_SIZE 256



/*
Given a string, check whether it meets the quality rules for
UMD Directory passwords

Implement:
   * A password must be at least 8 and no more than 32 characters
     in length.
   * A password must contain at least one uppercase letter.
   * A password must contain at least one lowercase letter.
   * A password must contain at least one character
     from the set of digits or punctuation characters
     (such as # @ $ & among others).
   * A password may not begin or end with the space character.
   * A password may not contain more than two consecutive identical
     characters.

Do not implement
   * A password may not be (or be a variation of) a dictionary word
     in English or many other languages. This includes making simple
     substitutions of digits or punctuation that resemble alphabetic
     characters (such as replacing the letter S in a common word with
     the $ symbol).
   * You may not reuse a password you have already used.
*/
int
check_password_rules(char s[])
{
}


int
```

```c
main ()
{
        char password[MAX_PASSWORD_SIZE];

        // Read password and check UMD rules

        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.