C Programming II
2020 Spring
Homework 01

Instructor: Po-Wen Chi

Due: 2020.03.24 PM 11:59

Policies:

- Zero tolerance for late submission.
- You need to prepare a README file about how to make and run your program. Moreover, you need to provide your name and your student ID in the README file.
- For the writing assignment, I only accept pdf. MS. doc/docx format is not acceptable. Moreover, please use Chinese instead of English.
- Do not forget your Makefile. For your convenience, each assignment needs only one Makefile.
- The executable programs should be hw0101, hw0102 ....
- You should pack your homework in one zip file. The file name should be StudentId_hw01.zip.

1 My String Library (20 pts)

In this class, I have shown you some standard string functions. Of course, you should use them when coding. However, sometimes you may need some functions that are not included in the standard string library. Do not worry, you can implement on your own. For your practice, I want you to implement some existing string functions in your own way.

```
char *mystrchr(const char *s, int c);
char *mystrrchr(const char *s, int c);
size_t mystrspn(const char *s, const char *accept);
size_t mystrcspn(const char *s, const char *reject);
char *mystrpbrk(const char *s, const char *accept);
char *mystrstr(const char *haystack, const char *needle);
char *mystrtok(char *str, const char *delim);
```

The usage of these functions should be the same with the standard version, including their return values. You need to prepare **mystring.h** and TA will prepare **hw0101.c**. Of course, Makefile is your own business. I prepare an example code for you, but it will not be the TA test file.

2 MAC Filter (20 pts)

In this class, you have practiced how to write a program to verify if a given string is a valid IPv4 address. This time, I want you to develop a program to check if a given string is a valid MAC address.

A media access control address (MAC address) is a unique identifier assigned to a network interface controller (NIC) for use as a network address in communications within a network segment. MAC Address is a 12-digit hexadecimal number (6-Byte binary number), which is mostly represented by Colon-Hexadecimal notation. You can use **ifconfig** to get the MAC address of your network card. An example is as follows:

- 88:d7:f6:54:15:0a

Of course, your program should accept both the upper case and the lower case. The string **88:d7:f6:54:15:0a:aa** is definitely invalid. If a string is a valid MAC address, you should print the MAC address with the hypen-hexadecimal notation. Otherwise, print **invalid**.

```
$ ./hw0102
Please enter the string: 88:d7:f6:54:15:0a
88-d7-f6-54-15-0a
$ ./hw0102
Please enter the string: 88:d7:f6:54:15:0a:22
invalid
```

3 Wildcard Matching (20 pts)

Undoubtedly, C standard string library provides some string matching functions, like **strcmp, strstr**. However, sometimes we want a string matching function that supports **patterns** instead of exactly words. Now I introduce a pattern called **wildcard**. First, you need to learn two symbols:

- **?**: Matches any single character.
- *****: Matches any sequence of characters (including the empty sequence).

For example, given a pattern **a?e**:

- **ae** does not match the pattern.
- **ace** matches the pattern.
- **ache** does not match the pattern.

For example, given a pattern **a*e**:

- **ae** matches the pattern.
- **ace** matches the pattern.
- **ache** matches the pattern.
- **apple** matches the pattern.

Now, given a pattern and an input string, please find all words that match the pattern. For your convenience, all test strings are composed of English lowercase alphabets only. The string is less than 512 bytes.

```
$ ./hw0103
Please enter the pattern: adam
Please enter the string: madam I am adam
Result: adam
$ ./hw0103
Please enter the pattern: ?adam
Please enter the string: madam I am adam
Result: madam
$ ./hw0103
Please enter the pattern: *a*m
Please enter the string: madam I am adam
Result: madam am adam
```

Hint: you can use some tricks that you have learned from your homework last semester.

4 Rational Number Arithmetic (20 pts)

In mathematics, a rational number is a number that can be expressed as the quotient or fraction $\frac{p}{q}$ of two integers, a numerator p and a non-zero denominator q.

Please develop a structure for the rational number and implement some arithmetic operations. You need to provide a header file **rational.h** and its .c file. TA will prepare hw0104.c which include your header file. Note that you should prepare a Makefile.

```
struct rational {
};

int rational_set( struct rational *r, int32_t p, int32_t q);
// return -1 if invalid; otherwise, return 0.
int rational_print( const struct rational r);
// in the form of (p,q)
void rational_add( struct rational *r, const struct rational r1, const struct rational r2);
// r = r1 + r2
void rational_sub( struct rational *r, const struct rational r1, const struct rational r2);
// r = r1 - r2
void rational_mul( struct rational *r, const struct rational r1, const struct rational r2);
```

```
// r = r1 * r2
void rational_div( struct rational *r, const struct rational r1, const struct rational r2);
// r = r1 / r2
```

Note that all rational number should be in the irreducible form.

5 Rational Number Arithmetic Part 2 (20 pts)

Now, please write a program to calculate the result of a given equation. For example, if a user wants to get the result of the following equation:

$$\frac{1}{2}+\frac{5}{6}\times\frac{3}{10}$$

```
$ ./hw0105
Please enter the equation: (1,2) + (5, 6)*(3,10)
(3,4)
```

Of course, you must follow arithmetic operation precedence. For simplicity, you do not need to consider parentheses.

6 fgets (5 pts)

fgets needs the programmer to give the buffer size. The problem is that the programmer does not know the user input size. How to solve this problem?Please write **Wildcard Matching** again without the string size limitation. You also need to describe your idea in your readme file.