National Taiwan Normal University
CSIE Programming II

Instructor: Po-Wen Chi
Due Date: PM11:59, March 16, 2021

Assignment 1

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 except foreign students.
- 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.

1.1 Hex to String (20 pts)

Please develop a program for a user to input a hex string and print its string. The hex string should be ended with 00.

```
$ ./hw0101
Please enter the hex string: 48656C6c6f20576F726C6400
Hello World
```

Note that all input characters must be **printable**. If the input hex string is invalid, please print a warning message.

1.2 Replacement (20 pts)

Replacement is a very important tool in a text editor. Please implement this function.

The user should input three strings. The first string is the full text, which contains at most 1024 characters. The second string is the keyword string that will be replaced with the last input string. The last two strings contain at most 128 characters. Please output the original text and the modified text. The old word should be colored with blue and the new word should be colored with red.

```
$ ./hw0102
Please enter the original text:
To be or not to be, that is the question.
Keyword:
be
New word:
do

Before:
To be or not to be, that is the question.
After:
To do or not to do, that is the question.
```

Note that I do not handle the color issue in this example but you need to do it.

1.3 Swap Strings (20 pts)

I believe that it is very simple for you to write a function that swaps two integers. This time, I want you to write a function that swaps two strings. You need to design this function yourself.

TA will prepare **hw0103.h** for you and you need to include it in your hw0103.c. hw0103.h will be as follows:

```
char *pStr01 = "Hello";
char *pStr02 = "World";

void print_answers()
{
    printf( "String 01:\n" );
    printf( "%s\n", pStr01 );
    printf( "String 02:\n" );
    printf( "%s\n", pStr02 );
    return;
}
```

You need to call TA's print function in your code. Of course, the string content will be different in TA's test file. The output should be

```
$ ./hw0103
String 01:
World
String 02:
Hello
```

Do not do silly things like print strings directly. Our TAs are smart enough to catch this cheating event.

1.4 JSON Reader (20 pts)

JSON is a popular text format for information exchange. You can see the following link to study this format.

https://zh.wikipedia.org/wiki/JSON

Now, I want you to develop a JSON reader. The user can input a JSON string which contains at most 2048 characters. Then the user will give a key and you need to return its value.

```
$ ./hw0104
Please enter the JSON string:
{ "firstName": "John", "lastName": "Smith", "sex": "male", "age": 25, "address": { "streetAddress": "21 2nd Street", "city": "New York", "state": "NY", "postalCode": "10021" }, "phoneNumber": [ { "type": "home", "number": "212 555-1234" },{ "type": "fax", "number": "646 555-4567" } ] }

Choice (0:Exit,1:Get) : 1
Key: firstName
Value: John
Choice (0:Exit,1:Get) : 1
Key: address.city
Value: New York
Choice (0:Exit,1:Get) : 1
Key: phoneNumber[0].number
Value: 212 555-1234
Choice (0:Exit,1:Get) : 0
Bye
```

For your simplicity, you can

- treat **space** as the only white space, which means the input string will not contain **newline** except the last character.
- skip the JSON format check.
- ignore the case that the key has **dot**.

BTW, you cannot use any existing JSON handling code.

1.5 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 **hw0105.c**. Of course, Makefile is your own business. Do not forget to make **hw0105.c** in your Makefile. I prepare an example code for you, but it will not be the TA test file.

1.6 Bonus: perror (5 pts)

In this class, I have shown you how to use a useful function called **strerror**. Actually, there is another similar function called **perror**. Please check the manual and see how to use it. Describe the difference between **strerror** and **perror**. Also provide an example code for **perror**.