C Programming II
2020 Spring
Homework 04

Instructor: Po-Wen Chi

Due: 2020.05.26 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 hw0401, hw0402 ....
- You should pack your homework in one zip file. The file name should be StudentId_hw04.zip.

1 Simple AWK (25 pts)

AWK is a domain-specific language designed for text processing and typically used as a data extraction and reporting tool. It is a standard feature of most Unix-like operating systems. AWK was initially developed in 1977 by Alfred Aho (author of egrep), Peter J. Weinberger (who worked on tiny relational databases), and Brian Kernighan; it takes its name from their respective initials.

Awk is a useful tool to extract wanted data from formated sources. I will give you a short demonstration in the class.

Now I want you to develop AWK ... of course not. I want you to develop a simple version. Given a file as **test.txt** follows.

```
Alice 70 60 50 40
Bob 100 101 102 103
Cathy 1 2 3 4
```

Your program should work in the following way.

```
./myawk '$1' '-->' '$3' test.txt
Alice-->60
Bob-->101
Cathy-->2
./myawk '$4' '+++' '$2' test.txt
50+++70
102+++100
3+++1
```

$1 and $3 are used to the first and the third fields which are separated by blanks. We use **quotation mark** to present the string.

You also need to support **-F** option to set **one character separator**. The default separator is blank. The option **-F** must be put after **myawk**. The last argument must be the file name.

2 JPEG File Interchange Format Thumbnail (25 pts)

The JPEG format is as follows. Do not worry, I will not ask you to decode it. Note that a JPEG can embed a thumbnail. Please write a program to embed (overwrite) a BMP thumbnail into a JPG and extract a thumbnail from a JPG into a BMP.

https://zh.wikipedia.org/zh-tw/JPEG%E6%96%87%E4%BB%B6%E4%BA%A4%E6%8D%A2%E6%A0%BC%E5%BC%8F

```
./myjpgthumb [option] [jpg] [bmp]
 -w --write: (Over)Write the bmp file to the jpg file's thumbnail.
 -e --extract: Extract the jpg file's thumbnail to the bmp.
```

3 Image Steganography (25 pts)

Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video. In this problem, I want you to implement a simple image data hiding program.

The concept is simple. Given a bmp image file, where R, G, B are presented in 8 bits. To embed a secret data in the cover bmp file, you just replace the last significant bits with the secret data bit, as shown in figure 1. If you want more capacity, you can use the last two significant bits, as shown in figure 2. Since we do not want to affect the original image file too much, the embedding order is from LSB to MSB.

Now I want you to embed and extract a secret data in/from the cover bmp image file.

```
./img_hide [option] [cover_bmp] [secret_data]
 -w, --write: Write the secret data to the cover_bmp.
 -e, --extract: Extract the secret data from the cover_bmp to the secret_data.
 -b, --bits=N: use last N bits. N is from 1 to 8. The default N is 1.
```

There are two points here.

1. The secret data size should be recorded in the bmp header reserved field which is 4 bytes long.

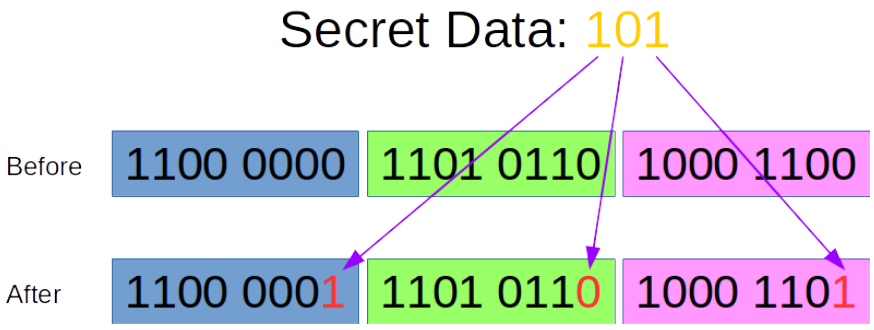

Figure 1: BMP LSB Steganography.

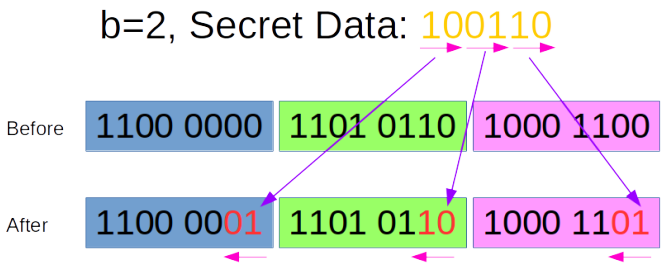

Figure 2: BMP Last Two LSBs Steganography.

2. If the secret data is larger than the cover bmp's hiding capacity, print an error message.

4 MACRO (25 pts)

Please define the following macros.

- FOR_EACH: to replace a for loop statement. The step is 1.
- FOR_EACH_STEP: to replace a for loop statement. The step is given by the user.
- SUM_OF_ARRAY: summation of all array elements.

All macros should support nested usage.
You should also provide an example code for these three macros.

5 Bonus: Autoconf (5 pts)

GNU Autoconf is a tool for producing configure scripts for building, installing and packaging software on computer systems where a Bourne shell is available.

Autoconf is agnostic about the programming languages used, but it is often used for projects using C, C++, Fortran, Fortran 77, Erlang or Objective-C.

A configure script configures a software package for installation on a particular target system. After running a series of tests on the target system, the configure script generates header files and a makefile from templates, thus customizing the software package for the target system. Together with Automake and Libtool, Autoconf forms the GNU Build System, which comprises several other tools, notably Autoheader.

Please write a small tutorial to show how it works.