C Programming II
2022 Spring
Homework 04

Instructor: Po-Wen Chi

Due: 2022.05.03 PM 11:59

Policies:

- **Zero tolerance** for late submission.
- **Plagiarism is not allowed.** Both source and copycat will be **zero**.
- 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.
  - Your Name and Your ID.
  - The functional description for each code.
  - Anything special.
- Please pack all your submissions in one zip file. **RAR is not allowed!!**
- For convenience, your executable programs must be named following the rule hwXXYY, where the red part is the homework number and the blue part is the problem number. For example, hw0102 is the executable program for homework #1 problem 2.
- I only accept **PDF**. MS Word is not allowed.
- Do not forget your Makefile. For convenience, each assignment needs only one Makefile.

1 Visual Cryptography (25 pts)

Visual cryptography is an interesting encryption technique. I think you all can learn how to do this with the following site.

`https://www.101computing.net/visual-cryptography/`

Now I want you to implement this visual cryptography method. The steps are as follows:

1. Given a colorful BMP image, make it gray scale only.

2. **Use 128 as the threshold to make the pixel black or white.**

3. Resize this image to 2×2 larger than the original image. That is, both width and height are twice than the original. So you can use 4 pixels to represent 1 pixel in the original image.

4. Use the method described in the above site to generate layer1, layer2 and overlap.

Note that the method described above only support black color and white color. In this problem, you can just use the gray scale. That is, each pixel can be gray.

```
$ ./hw0401 maldives.bmp
$ ls
maldives.bmp maldives_layer1.bmp maldives_layer2.bmp maldives_overlap.bmp
```

2 Code Generator (25 pts)

Coding is really annoying, especially Prof. Chi's assignments!! Please calm down. I have a good idea for you. How about writing a code to generate codes? What a genius proposal!!

Please see the following text file.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This is IPv4 header, which you will learn in the computer networking class. This is copied from RFC 791. RFC, request for comments, is a publication in a series, from the principal technical development and standards-setting bodies for the Internet, most prominently the Internet Engineering Task Force (IETF). Don't you think that this is well structured? So I want you to generate codes for this.

```
$ ./hw0402 -i ip.txt -n IPH -p ip
$ ls
ip.txt ip.c ip.h
```

The options are as follows:

- **-i**: Mandatory. The argument is the text file name which contains the structure.
- **-n**: Option. The argument is the structure name. Default: Test.

- **-p**: Option. The argument is the output file name. Default: test.

The generated **ip.h** and **ip.c** are as follows.

```
#pragma once

#include <stdint.h>

typedef struct _sIPH
{
  uint8_t version;
  uint8_t ihl;
  uint8_t typeofservice;
  uint16_t totallength;
  uint16_t identification;
  uint8_t flags;
  uint16_t fragmentoffset;
  uint8_t timetolive;
  uint8_t protocol;
  uint16_t headerchecksum;
  uint32_t sourceaddress;
  uint32_t destinationaddress;
  uint8_t options[3];
  uint8_t padding;
}IPH;

IPH * iph_init( void );
void iph_free( IPH * );
int iph_encode( void *, const IPH * );
int iph_decode( const void *, IPH * );
```

ip.c are implementations of these functions.
Some notes:

- You should use **suitable** type for each member. If the size is less than 8-bits, use uint8_t directly.
- If the member is not 8-bits, 16-bits, 32-bits and 64-bits, **use minimum required bytes number and represent it as an array.**
- **init** is to allocate enough memory.
- **free** is to free allocated memory.
- **encode** is to encode a structure to a memory buffer. Note that the output buffer must follow the given format. That is, version is 4 bits, not 1 byte. For your simplicity, **the prepared buffer size is definitely enough.** You only need to check NULL pointer.
- **decode** is to decode a memory buffer to a structure. Note that the input buffer must follow the given format. That is, version is 4 bits, not 1 byte.
- The generated code **must be compilable without any warnings**.

3 Source Code Highlighter (25 pts)

One of the most important feature is code highlighting. The editor can use different colors to present keywords and variables. Besides, it provides line numbers. Now it is your turn to develop a code highlighter. You all know how to print colorful words, right?

```
# ./hw0403 [OPTION] [FILE]
```

Options are as follows:

- Without any options, you simply display the whole source code.
- -n –linenum: Display the line number before each line.
- -c –color: Display the code and make each keyword colorful.
- -l –language: The given programming language. **Default is C**.

For simplicity, we use the following site as reference and you need to support these languages.

`https://github.com/e3b0c442/keywords`

You can choose any colors you like.

4 Game Cheater: Princess Maker 2 (25 pts)

Princess Maker 2 is a series of social simulation games where the player must act as a parental figure and raise a young girl. The second game in the series, Princess Maker 2, was released in 1993. The player takes role of a war hero who raises a girl to the age of 18. At the end of the game, the daughter goes into a line of work; what this work is, how much talent she has for it, her marital life, and her overall happiness all depend on the player's actions throughout the game.

The daughter has a set of statistics that fluctuate depending on the activities the player assigns to her schedule, including various part-time jobs, schooling, adventure, and free time. It is these statistics that ultimately determine her final occupation, her skill level in her line of work, and her overall happiness in life; her marital fate is also decided, in part, by these statistics. The game also makes use of some invisible statistics to determine the occurrence of certain special events.[citation needed] The player can converse with the girl and she will refer to the player as "father".

You can download the game from the following URL:

`https://dos.zczc.cz/games/%E7%BE%8E%E5%B0%91%E5%A5%B3%E6%A2%A6%E5%B7%A5%E5%8E%822/download`

Unfortunately, as your teacher, I do not have enough time to enjoy this game. So I want to cheat!! Figure 1 is your daughter's abilities and some parameters. Please develop a program to modify the daughter's attributes, health status, money, items stored in the save file. The interface is designed by yourself. The program name must be **hw0404**.

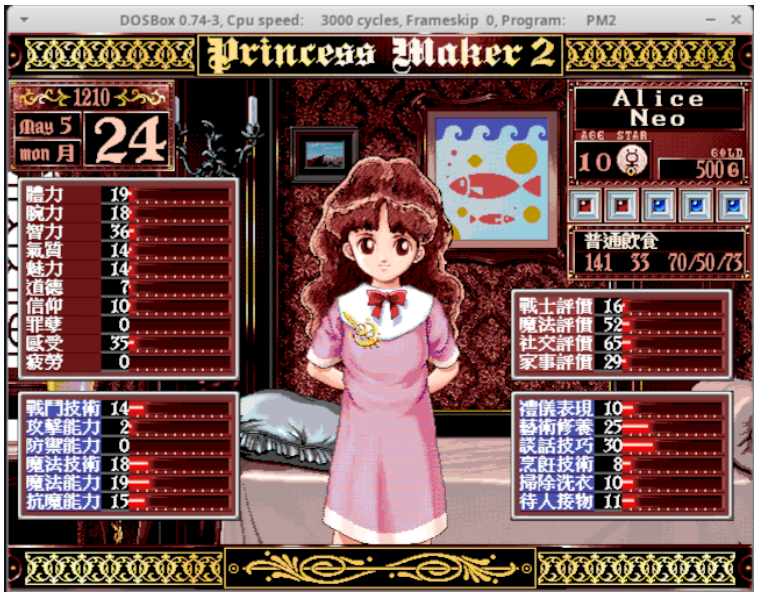

Figure 1: Your daughter's statistics.

5 Bonus: Wildcard (5 pts)

In this class, I have show you how to implement a process supporting lots of options and inputs. I believe that you all know how to do this. However, look the following scenario:

```
$ rm *.c
```

Suppose **rm** will receive only one argument ***.c** and there is no file called *.c. What happens here? Please describe how your computer works and design a lab to prove your description.