

CSE102 – Computer Programming

Homework #4

Due Date: 30/03/2020

Hand in: A student with number 20180000001 should hand in a 'c' file named 20180000001.c for this homework.

P.S.: Do not use arrays for any part of this homework!

Part 1. [25 pts] Welcome, our precious computer engineers. We were sniffing enemies' communications, and finally one of our agents capture a folder which contains many files. The files consist of numbers from 0 to 6 and mean nothing in this form but we know there are classified messages in these files, and we need you to decrypt and visualize the files for reading that messages. After perform much effort on the file, our experts notice that the numbers represents some characters such as 0, 1, 2, 3, 4, 5, 6 are ' ', '-', '_', '|', '/', '\', 'O' respectively (The first character is space character). You are asked to write a function that reads such files, decrypt them and visualize the secret messages in these files. Use encrypted_p1.img file to test your decryption function. The function prototype has given below:

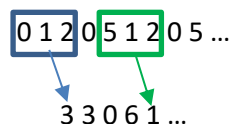
```
void decrypt_and_print (char* file_path)
```

Use a helper function for decryption of characters and do it with switch – case structure.

```
char decrypt_numbers (int number)
```

Part 2. [20 pts] Well done! We are appreciating you about reading the first secret message in the files, but we are not done yet. Some files don't mean anything when we use your decryption function in part 1. These files may be junk and inserted on purpose to prevent other files become decrypted. On the other hand, the files still seem suspicious to our stenographers. They say that there could be another message if the numbers in the file is replaced with the numbers gained from the formula:

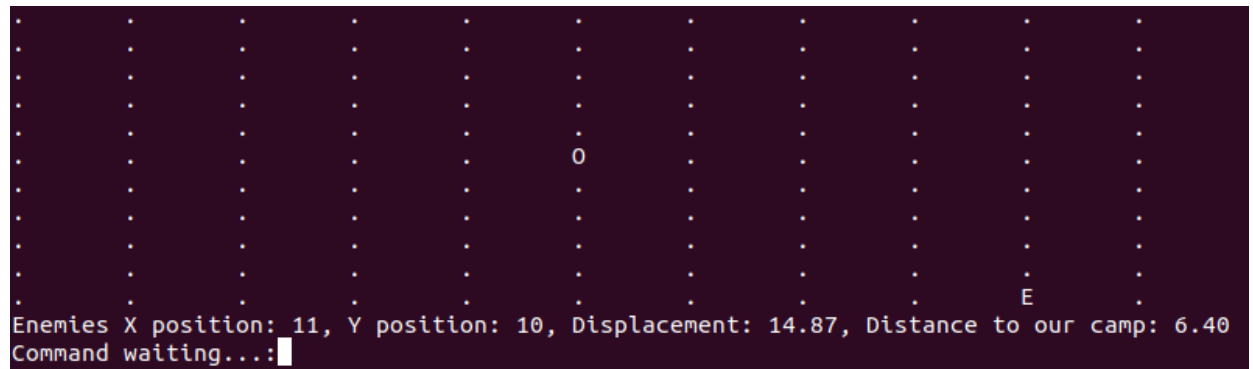
$$Number_i = \left(\sum_{j=0}^2 Number_{i+j} \right) \% 7$$



You are asked again to write a function that prints that deep secret messages in such files. By the formula, if there are no longer available three number for any position such as end of lines, you can assume the value of missing numbers are 0. Use encrypted_p2.img file to test your decryption function. **Don't change the original file, just apply the formula when you print the actual character.** The function prototype has given below:

```
void deep_decrypt_and_print (char* file_path)
```

Part 3. [25 pts] Perfect! Now, we know all the plans of enemies. Unfortunately, also the opponents know that we know their plans. Therefore, they are constantly changing the position of their camps. We want to make a machine to track the position of the enemy camp. The machine should have an interface that is shown below:



The O is our position and it is constant. E represents the enemies' current position. X and Y is the current axes of the position of the enemy camp. R shows the distance between current position and the last position of the enemy camp. D is the distance of enemy to our camp. The machine should work until the operator gives 'E' command to machine and it have to refresh the view of screen on each 'R' command. The prototype of the function of that machine is shown below:

```
void track_machine ()
```

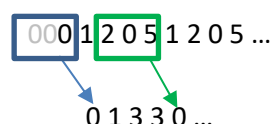
You have to refresh parameters by another function. The function prototype is:

```
void refresh_position (int *X, int *Y, double *D, double *R)
```

The battlefield consists of 11x11 positions and our camps is on the position (6, 6). Initial position of enemy is (0, 0) and next positions of the enemy camp should be generated randomly.

Part 4. [20 pts] Our tracking machine works like a charm! There is only one thing left to do. We want to manipulate the enemies' plans by leaking our messages into their network. The opponents use a new encryption algorithm that uses the formula below:

$$Number_i = \left(\sum_{j=0}^2 Number_{i-j} \right) \% 7$$



Therefore, you are asked to write a function that takes a message string as input and encrypts the message by using the formula below. The function saves the encrypted message in encrypted_p4.img file. Use decrypted_p4.img file for encryption. Use zeros as much as you need when encoding the first two numbers.

```
void encrypt_messages (char* file_path)
```

Part 5. [10 pts] Write a menu function to operate all the three parts. The menu should work forever until the operator selects the exit option. Your menu should seem as the figure below:

```
1-) Decrypt and print encrypted_p1.img
2-) Decrypt and print encrypted_p2.img
3-) Switch on the tracking machine
4-) Encrypt the message
5-) Switch off

Please make your choice:
█
```

The menu have to be in a function and your main function must contains only the menu function (and may be SRAND() function). The function prototype is:

```
void menu ()
```

General Rules:

1. Obey and do not break the function prototypes that are shown on each part, otherwise, you will get zero from the related part.
2. The program must be developed on Linux based OS and must be compiled with GCC compiler, any problem which rises due to using another OS or compiler won't be tolerated.
3. Note that if any part of your program is not working as expected, then you can get zero from the related part, even it's working in some way.
4. Upload your .zip file on to Moodle to deliver your homework. The zip file must consist of three .c file that contains your solutions. Name format can be found on the top of this homework sheet.
5. You can ask any question about the homework by sending an email to b.koca@gtu.edu.tr or by using the forum in the Moodle page of the course.