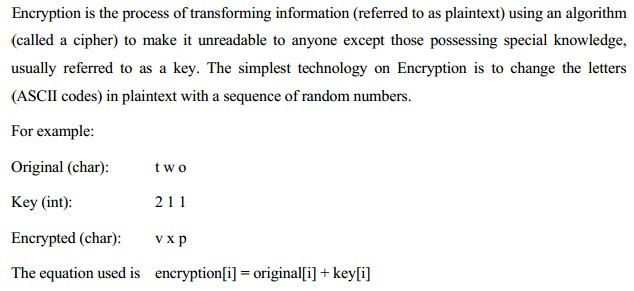
**Assignment6**

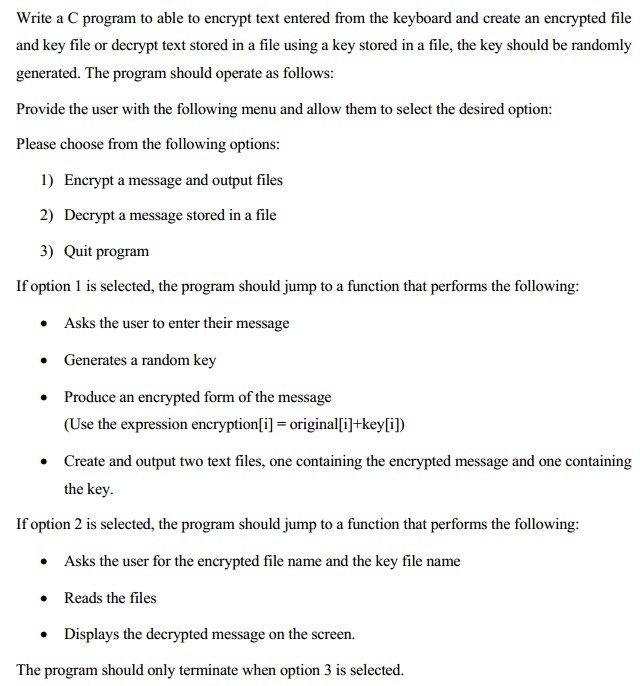
**1301058**

**Zhang Junming**

**exercise 1**

Question





Model Answer

Software Development Process

1. **Problem statement**

Write a C program, ask user to choose functions, one is input a paragraph of original strings, and then encrypt them to encrypted strings and output the encrypted file and key file. Another is lode encrypted and key file, and then decrypt them display the original strings.

1. **Analysis**

Inputs:

1. A paragraph of original strings.
2. Encrypted and key file name.

Outputs:

1. Encrypted and key file.
2. Decrypted message on the screen.

Additional requirements or constraint

ASCII table is useful.

**3. Design**

Algorithm

1. Adding ‘stdio.h’ ,‘stdlib.h’,’time.h’ and ’string.h’ library.
2. Define ‘string\_length is 5000.
3. Create first defined function ‘menu ()’:

(1) Display the main menu.

(2) Ask user to input number to choose functions.

1. Create second defined function, ‘option1():
2. char file1 and file2 – represent Encrypted and Key file name.
3. int i - for counting.
4. int length - represents the length of strings.
5. Char Origin[string\_length] – represents origin strings.
6. Char Key[string\_length] – represents Key strings.
7. Char Encrypted[string\_length] – represents Encrypted strings.
8. FILE \*fp1 and \*fp2-represents Encrypted and Key file.
9. Ask user to enter original strings.
10. Read and store these strings in **Origin**.

(10) Counting how many characters in **Origin** and store in **length**.

(11) Resetting time.

(12) Setting up a loop use ‘for’ ,create **key** equal to random number and **Encrypted** equal to **Origin** plus **Key** (Encrypted[i] = Origin[i] + Key[i]).

(13) Ask user to enter address and name of Encrypted file and display standard format.

(14) Read and store file name in **file1**.

(15) Create and open Encrypted file.

(16) Setting up a loop use ‘for’, input information to text which store in **Encrypted**.

(17) Close Encrypted file.

(18) Ask user to enter address and name of Key file and display standard format.

(19) Read and store file name in **file2**.

(20) Create and open Key file.

(21) Setting up a loop use ‘for’, input information to text which store in **Key**.

(22) Close Key file.

(23) Use system ‘pause’ and ‘cls’ clear screen after using this defined function.

5.  Create third defined function, ‘option2():

1. char file1 and file2 – represent Encrypted and Key file name.
2. int i - for counting.
3. int length - represents the length of strings.
4. Char Origin[string\_length] – represents origin strings.
5. Char Key[string\_length] – represents Key strings.
6. Char Encrypted[string\_length] – represents Encrypted strings.
7. FILE \*fp1 and \*fp2-represents Encrypted and Key file.
8. Ask user to enter Encrypted file’s name file and display standard format.
9. Read and store file name in **file1**.

(10) Search and open this file.

(11) If cannot open this file, display can’t open this file. Return start.

(12) If open this file successful, display enter data successful.

(13) Ask user to enter Key file’s name file and display standard format.

(14) Read and store file name in **file2**.

(15) Search and open this file.

(16) If cannot open this file, display can’t open this file. Return start.

(17) If open this file successful, display enter data successful.

(18) Setting up a loop use ‘for’, lode these information in **Encrypted** and **Key**. Let **Origin** equal to **Encrypted** minus **Key** (Origin[i] = Encrypted[i]-Key[i]).

(19) Close Encrypted and Key file.

(20) Use system ‘pause’ and ‘cls’ clear screen after using this defined function.

6. int ch – for receive EHTER.

int choose – represents options.

7. Setting up a loop use ‘do while’, ask user to input a number choose functions, only input ‘3’ could out this loop.

8. Using defined function menu().

9. Read and store entered value in **choose**.

10. Receive EHTER.

11. Setting up ‘switch’ function of **choose**.

Case 1: using option1 defined function.

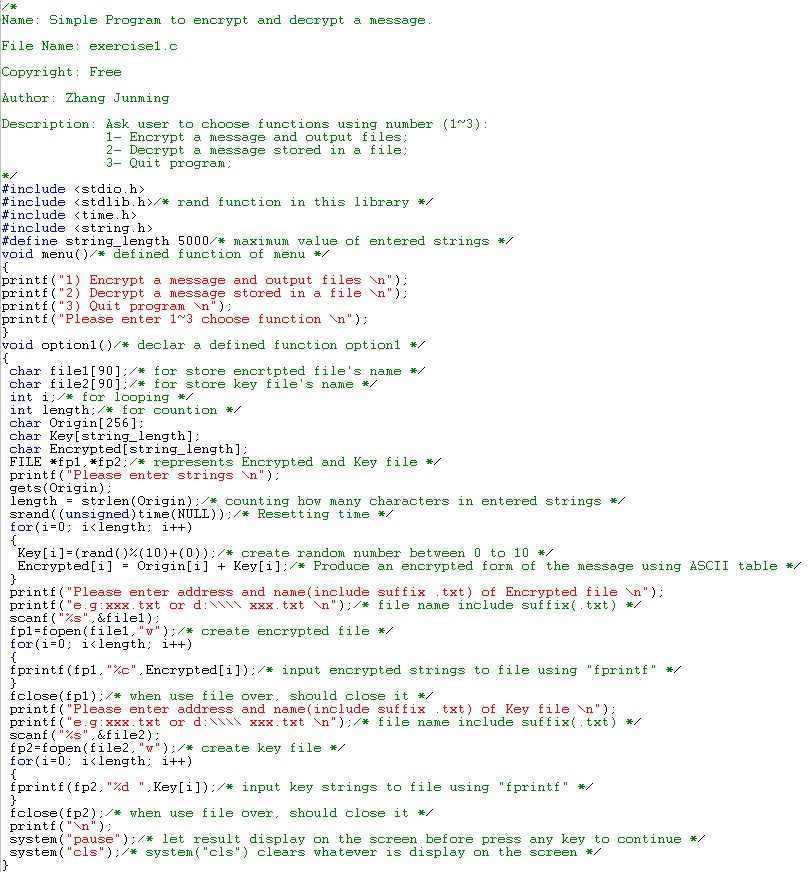
Case 2: using option2 defined function.

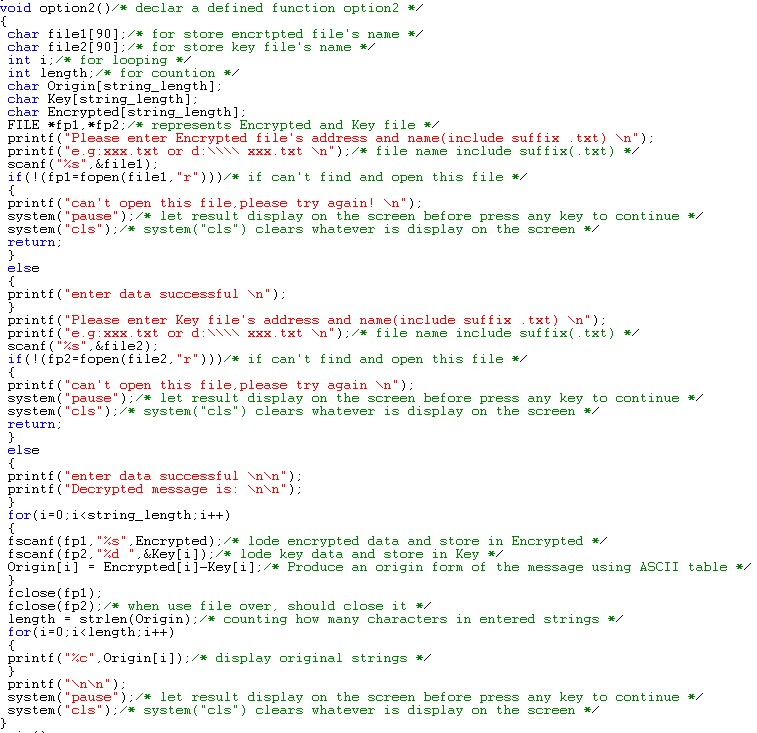
Case 3: noting.

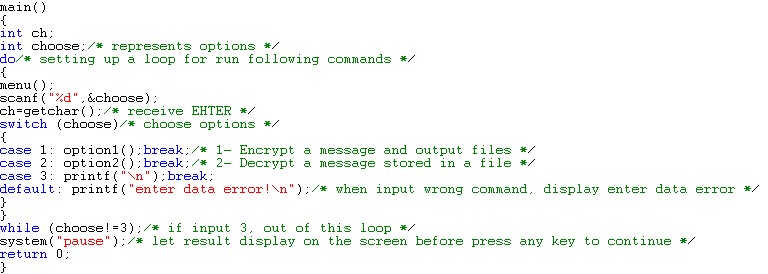
Default when entered number is not in 1~3, display enter data error.

1. **Implementation**:

See the C code in file exercise1.c with comments.





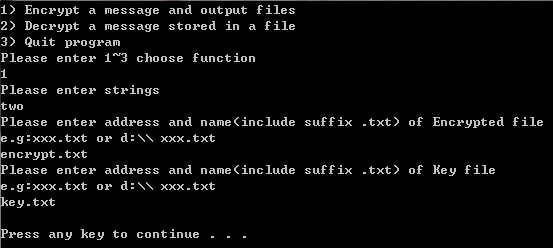


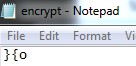
1. **Testing:**

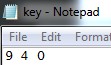
The C program was tested by carrying out a set of experiments; and the C program output was verified successfully. For instance,

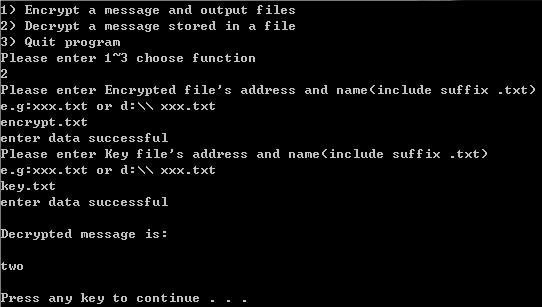
Test 1



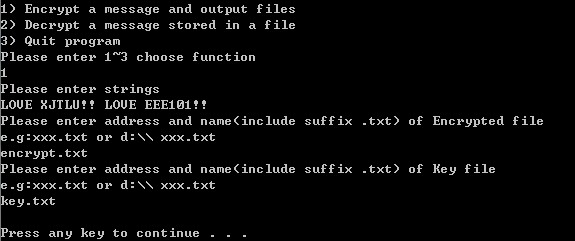


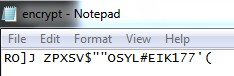


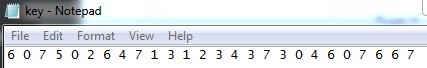




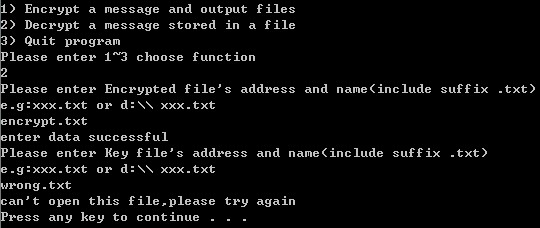
Test 2







When enter wrong text:



When enter right text:

