SWIN
BUR
NE

SWINBURNE
UNIVERSITY OF
TECHNOLOGY

COS10007 -Developing Technical Software

Assignment 2

(This assignment is worth 25% of the subject's total assessment marks)

Due Date: Sunday 16th December 2018 at 11:59 pm

You may be asked to demonstrate/explain your work to the tutor, if you are absent/unavailable or fail to demonstrate properly, zero marks will be awarded.

Please note, this is an individual task and it will be checked for plagiarism. All the involved parties will be penalised if any plagiarism is found.

Please visit https://goo.gl/hQ87zq for more details.

Instructions

1. This assignment contains 3 questions. Q1 is for 14 marks, Q2 is for 5.5 marks, Q3 is for 5.5 marks. The total assignment is for 25 marks and see the detailed rubric available on canvas
2. Submit word document with all your code pasted and also the screenshots
3. You should also submit your .c and .h files along with the document. Zip both of them together
4. Use only .doc, .docx extensions – no other format will be accepted for marking
5. Marks will be given for proper indentation and comments
6. Assignment Demonstration is mandatory

Question 1:

Assume you are working as a programmer in a communication field. Your task is to alter the original data before transmitting because of a security reason. All of their data is an integer that contains the number of digits between two and eight (inclusive), thus the possible integers are between 10 and 99999999. You have to read in an integer and complete the following stages.
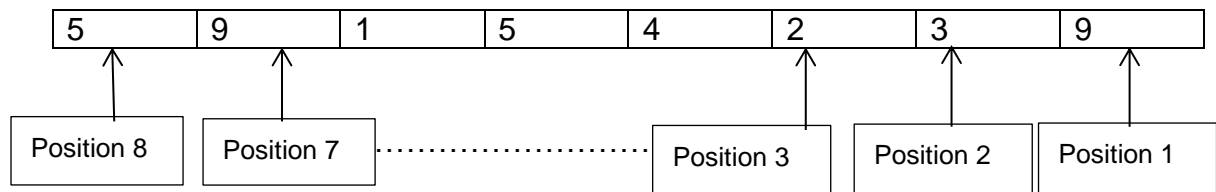
Stage 1:
Your task is to modify the data for transmission according to the following set of rules.

1. If the number is a two digit number you swap the positions of the numbers (eg: if the number is 56, altered number is 65).
2. If the number is a three digit number, digits at positions 1 and 3 are swapped. (eg: number is 123, altered number is 321)
3. If the number is four digits or above the following rules apply.

   Replace the first digit by the remainder after the sum of that digit plus 1 is divided by 10, the second digit by the remainder after the sum of that digit plus 2 is divided by 10, third digit by the remainder after the sum of that digit plus 3 is divided by 10, fourth digit by the remainder after the sum of that digit plus 4 is divided by 10 and so on.

Position of the number is counted from right to left.

5	9	1	5	4	2	3	9

Position 8	Position 7	. . . .	Position 3	Position 2	Position 1

Stage 2: Extra protection (done only on 8 digits numbers – if the user chooses this option)

Eight digit data contains more valuable information, so you are asked to enable extra protection if the user chooses to.

The extra protection is done on **modified number in stage one** by swapping the digit in position 1 with digit in position 8, digit in position 2 with digit in position 7, digit in position 3 with digit in position 6, digit in position 4 with digit in position 5.

If the modified number in stage 1 is as shown below:

2	3	4	5	6	7	8	9

The number after stage 2 will look like:

9	8	7	6	5	4	3	2

Stage 3: (Decoding)

Once the encoding stages are over, write the codes to recover your original data from the encoded data.

Other requirements:
- This assignment must be written in C programming. The program should use multiple modules i.e., more than one .c file and .h header
- Your code must have appropriate header(multiline/block) comments including your name and student number, the name of the .c file, the purpose of the program, brief explanations of variables and explanations of any code, which is not obvious to another programmer, summarising the input, output and local variables as well as expressions used in your program and test data.
- Include inline (single-line) comments throughout the program describing important statements.
- Use appropriate and descriptive variable following the naming rules and conventions.
- Write a brief (no more than two pages) report, which illustrates your program design (algorithm or flowchart, identification of variables, constants) and

include evidence of testing – screen shots or pasted output text of several tests, and the contents of the .c file, .h file

- Marks will be allocated depending on the amount of original work submitted. Marks will be deducted for plagiarised and/or un-attributed work.

Screenshots showing working program:

```
Enter a number
8
Not a valid input, re enter the number
-10
Not a valid input, re enter the number
67
Number contains 2 digits

The encoded number is 76

The decoded number is 67
```

```
Enter a number
230
Number contains 3 digits

The encoded number is 032

The decoded number is 230
```

```
$ assign1
Enter a number
9876
Number contains 4 digits

The encoded number is 3197

The decoded number is 9876
```

```
Enter a number
50000
Number contains 5 digits

The encoded number is 04321

The decoded number is 50000
```

```
Enter a number
987654
Number contains 6 digits

The encoded number is 531975

The decoded number is 987654
```

```
Enter a number
34523450
Number contains 8 digits

Do you want to do second level of encryption, enter yes or no? yes
The encoded number is 17777111

The decoded number is 34523450
```

Question 2

Write a complete C++ program that use a random function to produce two positive one-digit integers (0 includes). The program should then prompt the user with a question, such as "How much is 2 + 3? "

The user then inputs the answer. Next, the program checks the user's answer. If it's correct, the program should randomly display any of the following messages
1. Very good!
2. Excellent!
3. Nice work!
4. Keep up the good work!
and ask another addition question.

If the answer is wrong, randomly display any of the following messages
1. No. Please try again.
2. Wrong. Try once more.
3. Don't give up!
4. No. Keep trying

and let the user try the same question repeatedly until the user finally gets it right. A separate function should be used to generate each new question. This function should be called once when the application begins execution and each time the user answers the question correctly.

Sample Screenshot:

```
How much is 1 + 7?
8
Keep up the good work!
How much is 2 + 0?
0
Wrong. Try once more
How much is 2 + 0?
0
No. Please try again.
How much is 2 + 0?
2
Keep up the good work!
How much is 7 + 1?
8
Excellent!
How much is 4 + 0?
4
Excellent!
How much is 8 + 4?
-1
```

Question 3

Write a complete C++ program to create a one-dimensional array to read 26 alphabetical letters (your program should be able to detect and print out an error message if a non-alphabetical letter is entered). As each letter is entered, print a message saying 'duplicate letter' if the letter is already in the array. Write a function that can sort the array after all 26 letters have been entered. Write another function that print out the most frequent letter and number of times it was entered. Prepare for the case where all 26 letters are different, or all are the same.

Sample Screenshots: