

Western University
Department of Computer Science
CS 1027 (Computer Science Fundamentals II)
Assignment 2

Professor: Ali Bou Nassif, Ph.D., P.Eng.

Instructions: Read carefully before you start.

This assignment should be done individually. The deadline is Wednesday July 15, 2015 at 11:54 pm. Please refer to the course outline (**Assignments Submissions and Late Assignments Policy Section**) for the late penalty. Each student must submit a zip file (zip and NOT rar) that compresses the main folder of his/her assignment (check the last paragraph of the assignment for more details). All files should work under Eclipse. It is very important that you comment your code thoroughly. Marks will be deducted if the code is not commented properly. The overall mark is 120 (20 bonus marks)

Question 1: 100 points

The goal of this question is to design an advanced version of Caesar Cipher (Read Unit 9 slides 12 to 16). In the original version of encoding a string using Caesar Cipher, each character is shifted by 'k' characters. For example, if 'k' = 2, the string "abc" will become "cde".

In this question, you are required to implement an advanced version of Caesar Cipher by shifting each character of the string by a certain number of characters using a repeating key (see slides 14 and 15 in Unit 9). In this assignment, we will assume that the key value is 12345 which will be used to encode the incoming string. When the key is exhausted, start over at the beginning of the key. For example, the string "aaaaaaaaaa" will be encoded to "bcdefbcdefb" using the key 12345.

Write a file called **encode.java**. You have to address the following requirements:

- 1- When you run your program, the user will be prompted to enter a string. The string might contain space characters. Check the skeleton of Assignment1 to see how you can input a string using Scanner object. The main difference between this assignment and Assignment1 is that in Assignment1 we used the method **Scanner.next()** because the inputted sting in Assignment1 does not contain space characters. In this assignment, space characters are allowed and **Scanner.next()** method will not work in this case, so you have to use a different method
- 2- The key 12345 should be stored in a Queue. The Queue should be implemented using LinkedList class which is part of java.util package. The LinkedList class is implementing the Queue Interface (check the last slide in Unit 9)
- 3- The inputted string should be parsed character by character (you can use a for loop with charAt() method as in assignment1). The encoded character should be stored in a linked list. This means, you should have two containers, one is a Queue that stores the key and one is a linked list that stores the encoded string
- 4- In order to encode a character, take its ASCII <http://www.asciitable.com/> code (the integer value of the character), then add it to its corresponding key value, then convert the new value to

a character, then store this character in the linked list. For example, if you wish to encode the message "Hello", take the ASCII code of 'H' which is 72, add it to 1 (the first digit of the key), this becomes 73. Convert this integer to its character value which is 'I', then insert 'I' to the linked list. You encoded the character 'H', now do the same for 'e'. Its ASCII code is 101, add it to 2 (the second digit of the key), this becomes 103. Convert this integer to its character value which is 'g', then insert 'g' to the linked list. Repeat until you encode the string "Hello" which will become "lgopt"

- 5- All space characters in the inputted string should be ignored during string parsing and should not be encoded. Note that the ASCII code of the space character is 32. In this case, the encoded string should not contain any space.
- 6- When you have finished parsing the inputted string and stored its encoded version in the linked list, print the elements of the list. Use a for loop to do so and do not use the default toString method.

Sample output of the program:

Input a string you wish to encode:

Hello, how are you?

Output string: lgopt-Jr{fsg|sz@

Question II (20 points)

Write a file called decode.java that takes an encoded string generated from Question I as input and decode it (converting it back to its original format but without space characters). The requirements of this question are the same as the requirements of Question 1 (the Queue stores the same key (12345) and the linked list stores the decoded string. The only difference is that the inputted string in this question contains no space characters.

Sample output of the program:

Input a string you wish to decode:

lgopt-Jr{fsg|sz@

Output string: Hello,howareyou?

Deliverables:

1. A zip file of name **your uwo user name**. This file should contain two folders named question1 and question2 (all lower case). The question1 folder should contain the file encode.java. The question2 folder should contain the file decode.java
2. ReadMe.pdf file. The submission of this file is optional. However, if you feel that you would like to tell the TAs any useful information about your assignment, feel free to write it in this file. This file can be added to the main folder
3. Submit the zip file to OWL before the deadline