NYU, Tandon School of Engineering
CS-UY 1114 Introduction to Programming and Problem Solving — Spring2018

**Homework #5**

**Submission instructions:**

1. You should submit your homework in the NYU Classes system.

2. For this assignment you should turn in 10 '.py' files, each containing a script for each section of each question. Name your files: 'YourNetID_hw5_q1.py', 'YourNetID_hw5_q2a.py', 'YourNetID_hw5_q2b.py', 'YourNetID_hw5_q3.py', 'YourNetID_hw5_q4.py', 'YourNetID_hw5_q5.py', 'YourNetID_hw5_q6.py', 'YourNetID_hw4_q7a.py', 'YourNetID_hw4_q7b.py', 'YourNetID_hw5_q8.py'.

**Question 1**

(From the textbook, page 219, question 4) Read from the user a string containing odd number of characters. Your program should:

   a) Print the middle character.
   b) Print the string up to but not including the middle character (i.e., the first half of the string).
   c) Print the string from the middle character to the end (not including the middle character).

Sample output:

Enter an odd length string: Fortune favors the bold

Middle character: o

First half: Fortune fav

Second half: rs the bold

## Question 2

Write **two versions** of a program that reads a character (string of length 1) from the user, and classifies it to one of the following: lower case letter, upper case letter, digit, or non-alphanumeric character

  a) In the first program, use string methods.
  b) In the second program, do not use string methods.

Sample output (4 different executions):

Enter a character: j
j is a lower case letter.

Enter a character: 7
7 is a digit.

Enter a character: ^
^ is a non-alphanumeric character.

Enter a character: C
C is an upper case letter.


## Question 3

Write a program that will read and evaluate a mathematical expression. The expression is of the form: ***oprand1 op operand2***, where operand1 and operand2 are positive integers and op is an operator, which is either +, -, * or /.
For example: 24 + 65 and 276 * 2 are legal expressions.
<u>Assumption</u>: There is a single space between each operand and the operator.

Sample output (2 different executions):

Enter a mathematical expression: 5 + 10
5 + 10 = 15

Enter a mathematical expression: 81 / 9
81 / 9 = 9

**Question 4:**

Write a program that reads an English word from the user, and prints how many vowels and how many consonants it contains.

Note: You may assume the letter Y is not a vowel.

Sample output (2 different executions):

Enter a word: test
test has 1 vowels and 3 consonants.

Enter a word: Apple
Apple has 2 vowels and 3 consonants.

**Question 5:**

The ACME Widgets Company has a new password policy. Passwords must be at least 8 characters long and must contain the following:

- At least two uppercase letters
- At least one lowercase letter
- At least two digits
- At least one special character: ! @ # $

Write a program that reads in a string and determines whether it is a valid password.

Sample output (2 different executions):

Enter a password: P4Ssword1!
P4Ssword1! is a valid password.

Enter a password: password
password is not a valid password.

**Question 6:**

Ask user to input a string containing only lower case letters. Determine if the input is ordered in a lexicographical increasing order.

For example, an execution would look like:

Please enter a string of lowercase letters: abgkp
abgkp is increasing.

Another execution would look like:

Please enter a string of lowercase letters: abgcp
abgcp is not increasing.

**Question 7:**

In this question, we will use a **simplified** version of the Roman Numerals System to represent positive integers.

The digits in this system are I, V, X, L, C, D and M. Each digit corresponds to a decimal value, as showed in the following table:

| Roman digit | I | V | X | L | C | D | M |
|---|---|---|---|---|---|---|---|
| Decimal value | 1 | 5 | 10 | 50 | 100 | 500 | 1000 |

A number in the *simplified Roman numerals system* is a sequence of Roman digits, which follow these 2 rules:

1. The digits form a monotonically non-increasing sequence. That is the value of each digit is less than or equal to the value of the digit that came before it.
   For example, DLXXVI is a monotonically non-increasing sequence of Roman digits, but XIV is **not**.
2. There is no limit on the number of times that 'M' can appear in the number.
   'D', 'L' and 'V' can each appear at most one time in the number.
   'C', 'X' and 'I' can each appear at most four times in the number.

For example: IIII, XVII and MMMMMMDCCLXXXXVII are legal numbers in our simplified Roman numeral system, but IIIII, XIV, VVI and CCXLIII are **not**.

Write 2 programs:

a. Reads from the user a number that is represented in the simplified Roman numerals system, and prints it's decimal value.

   Sample output:

   Enter number in the simplified Roman system:
   CXXXXVII

   CXXXXVII is 147

b. Reads from the user a (decimal) number, and prints it's representation in the simplified Roman numerals system.

   Sample output:
   Enter decimal number:
   147

   147 is CXXXXVII

**Question 8:**
Ask the user to input a line of text, and a character *ch*. Your program should:
1. **Create a string variable** that contains the text after removing all occurrences of the character *ch* from the input text.
2. Print that string.

For example, an execution would look like:
Please enter a line of text: This is a line of text.
Please enter the character you want to remove: s
Thi i a line of text.