

1. Write a Java program that will take a String as input and convert the lowercase letters to the uppercase letters. The program should keep the other characters (i.e. uppercase letters and symbols) unchanged. *[You are not allowed to use .toUpperCase() to solve the problem]*

| Sample Input | Output |
|----------------------|----------------------|
| JavaProgrammer123++ | JAVAPROGRAMMER123++ |
| I am a hungry coder! | I AM A HUNGRY CODER! |

2. Write a Java program to check if a string is palindrome or not. A string is palindrome if the reverse of the string is the same as the original string. *[You cannot use any built-in functions other than .equals(), .charAt(), .length()]*

| Given String | Output |
|--------------|--------|
| madam | true |
| racecar | true |
| Abracadabra | false |

3. Write a Java program that splits a given string on a given split character. The first input is a String and the second input is the character that will be used to split the first String.

| Sample Input | Sample Output |
|--------------------------------------|---|
| I-love-Java - | I love Java |
| tom@gmail,harry@yahoo,bob@gmail , | tom@gmail harry@yahoo bob@gmail Explanation: The second input which is the character ',' is used to split the first input string 'tom@gmail,harry@yahoo,bol@gmail,mary@gmail' into four separate email addresses. |

4. Write a JAVA program that will take one string input from the user. Then prints the string in reverse.

| Sample Input | Output |
|--------------|--------|
| ABCD | DCBA |
| Hello! | !olleH |

5. Captain Jack and his crew have discovered a treasure chest full of gold coins. However, the chest comes with a mysterious lock. To open it, they need to input a phrase that should contain a combination of characters where vowel count is divisible by 3 and consonant count is divisible by 5. Write a Java program to help Captain Jack determine if the input phrase has the correct number of vowels and consonants to unlock the treasure chest.

Note: Vowels and Consonants count has to be greater than 0 for the treasure to open

| Sample Input | Output |
|------------------|--|
| Yo-hoo-hoo! | Blimey! No Plunder!! Explanation: The input string has five vowels 'o, o, o, o, o' which is not divisible by 3 and the same applies for consonant rules as well where count of consonants is not divisible by 5. Hence the chest cannot be opened. |
| Yo-ho-Ya-ho-hoo! | Aaarr! Me Plunder!! Explanation: The input string has six vowel count which is divisible by 3 and the same applies for consonant rules as well where count of consonants is divisible by 5. Hence the chest can be opened. |
| aoouii-iii | Blimey! No Plunder!! |

| | |
|--|---|
| | Explanation: Here the vowel count is divisible by 3, but the consonant count being 0 resulted in the chest not being opened. |
|--|---|

6. Write a Java program that takes a string as input and reverses the order of words in it.

| Sample Input | Output |
|-------------------|-------------------|
| CSE110 is easy | easy is CSE110 |
| Attention please! | please! Attention |

7. Write a Java program that takes two strings (**lowercase**) as user input. Your task is to concat the two strings except the common characters present in the strings. Then print the modified string (uppercase). You can assume that each string will only contain unique characters; **that means there will be no duplicate characters in a particular string.**

| Sample Input | Output |
|---------------|--------|
| abcd bdgc | AG |
| arose rail | OSEIL |

8. Write a Java program that takes a sentence as user input and displays the sentence in aLtErNaTiNg CaPs format. Note that, you have to ignore spaces and other punctuations while altering the characters. Also, your new sentence will always start with lowercase letters.

| Sample Input | Output |
|--|---|
| Hello World! | hElLo WoRiD! |
| I love Java String Problems. Those are easy. | i LoVe JaVa StRiNg PrObLeMs. ThOsE aRe EaSy. |

9. Write a Java program that takes a string as input and determines whether the string qualifies as a strong password. A strong password is defined by the following criteria:

- The password must be at least 8 characters long.
- The password must contain characters at least one -
- Uppercase letters (A-Z)
- Lowercase letters (a-z)
- Digits (0-9 n)
- Special characters (e.g., !, @, #, \$, etc.)

Print **'True'** if the password is strong, and **'False'** otherwise. You may consider any character excluding the alphabet, digit, and space as special characters.

| Sample Input | Output |
|----------------|--------|
| StrongPass123! | True |
| Weak123 | False |

10. Write a Java program that takes a string as input and prints all the substrings of that string. Then count the number of substrings that starts and ends with same character.

Find Tracing tasks on the next page.....

10. Trace the following code, create a tracing table and write the outputs.

| | |
|----|---|
| 1 | public class HWTracing01 { |
| 2 | public static void main(String[] args) { |
| 3 | int x = 0, y = 0; |
| 4 | String sum = "0"; |
| 5 | double p; |
| 6 | while (x < 9) { |
| 7 | y = x / 2; |
| 8 | while (y < x) { |
| 9 | p = (x + 5.0) / 2; |
| 10 | sum = (sum + 2) + x + "y * 2" + (int) p ; |
| 11 | System.out.println(sum) ; |
| 12 | y = y + 1; |
| 13 | } |
| 14 | x = x + 2; |
| 15 | if (x > 5) { |
| 16 | sum = "2"; |
| 17 | } else { |
| 18 | sum += "3"; |
| 19 | } |
| 20 | } |
| 21 | } |
| 22 | } |

11. Trace the following code, create a tracing table and write the outputs.

| | |
|----|--|
| 1 | public class HWTracing02{ |
| 2 | public static void main(String[] args) { |
| 3 | int test = 1; |
| 4 | int j = 0, k = 100; |
| 5 | while (k > 0){ |
| 6 | while (j < k){ |
| 7 | test = k - j + 11; |
| 8 | System.out.println(1 + test / 3 +"12"); |
| 9 | j+=10; |
| 10 | } |
| 11 | k-=10; |
| 12 | } |
| 13 | } |
| 14 | } |