

Q1 Write a Java program to find the longest substring from a given string that doesn't contain any duplicate characters.

Example:

- Input : Welcome to PowerRouter.
- Output : Welcome

```
package package3;
import java.util.HashMap;
public class LongestSubstringWithoutDup {
    public static String findLongestSubstring(String input) {
        int n = input.length();
        int start = 0;
        int maxLength = 0;
        HashMap<Character, Integer> charIndexMap = new HashMap<>();
        for (int end = 0; end < n; end++) {
            char currentChar = input.charAt(end);
            if (charIndexMap.containsKey(currentChar)) {
                start = Math.max(charIndexMap.get(currentChar) + 1, start);
            }
            if (end - start + 1 > maxLength) {
                maxLength = end - start + 1;
            }
            charIndexMap.put(currentChar, end);
        }

        return input.substring(start, start + maxLength);
    }

    public static void main(String[] args) {
        String input = "Welcome to Power Router";
        String longestSubstring = findLongestSubstring(input);
        System.out.println("Input: " + input);
        System.out.println("Output: " + longestSubstring);
    }
}
```

```

cal_percentage.java  LongestSubstringWithoutDup... × FibonacciSeries.java  StringPermutations.java
1 package package3;
2
3 import java.util.HashMap;
4
5 public class LongestSubstringWithoutDup {
6
7     public static String findLongestSubstring(String input) {
8         int n = input.length();
9         int start = 0;
10        int maxLength = 0;
11        HashMap<Character, Integer> charIndexMap = new HashMap<>();
12
13        for (int end = 0; end < n; end++) {
14            char currentChar = input.charAt(end);
15
16            if (charIndexMap.containsKey(currentChar)) {
17                start = Math.max(charIndexMap.get(currentChar) + 1, start);
18            }
19
20            if (end - start + 1 > maxLength) {
21                maxLength = end - start + 1;
22            }
23
24            charIndexMap.put(currentChar, end);
25        }
26
27        return input.substring(start, start + maxLength);
28    }
29
30    public static void main(String[] args) {
31        String input = "Welcome to Power Router";
32        String longestSubstring = findLongestSubstring(input);
33        System.out.println("Input: " + input);
34        System.out.println("Output: " + longestSubstring);
35    }
36

```

Q2 Write a Program for the first 10 numbers of Fibonacci series.

Example:

○ Output : 0,1,1,2,3,5,8,13,21,34

```

package package3;
public class FibonacciSeries {
    public static void main(String[] args) {
        int n = 10; // Number of Fibonacci numbers to generate
        printFibonacciSeries(n);
    }
    public static void printFibonacciSeries(int n) {
        int first = 0, second = 1;

```

```

    System.out.print("Fibonacci Series (first " + n + " numbers): ");
    System.out.print(first + ", " + second);
    for (int i = 2; i < n; i++) {
        int next = first + second;
        System.out.print(", " + next);
        // Update values for the next iteration
        first = second;
        second = next;
    }
    System.out.println(); // Move to the next line after printing the series
}
}

```

The screenshot shows an IDE with several tabs open: `cal_percentage.java`, `LongestSubstringWithoutDu...`, `FibonacciSeries.java` (active), `StringPermutations.java`, `AnagramChecker.java`, and `StringReversal.java`. The `FibonacciSeries.java` file contains the following code:

```

1 package package3;
2
3 public class FibonacciSeries {
4
5     public static void main(String[] args) {
6         int n = 10; // Number of Fibonacci numbers to generate
7         printFibonacciSeries(n);
8     }
9
10    public static void printFibonacciSeries(int n) {
11        int first = 0, second = 1;
12
13        System.out.print("Fibonacci Series (first " + n + " numbers): ");
14        System.out.print(first + ", " + second);
15
16        for (int i = 2; i < n; i++) {
17            int next = first + second;
18            System.out.print(", " + next);
19
20            // Update values for the next iteration
21            first = second;
22            second = next;
23        }
24
25        System.out.println(); // Move to the next line after printing the series
26    }
27 }

```

The console output at the bottom shows the execution result:

```

<terminated> FibonacciSeries [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Dec 21, 2023, 9:47:18 PM - 9:47:18 PM) [pid: 7684]
Fibonacci Series (first 10 numbers): 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

```

Q3 Write a program to print all permutations of a string.

Example:

- Input : CAT
- Output : CAT , CTA , ACT , ATC , TAC , TCA

```
package package3;
import java.util.Scanner;
public class StringPermutations {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();
        System.out.println("Permutations of the string:");
        printPermutations(input);
        scanner.close();
    }
    public static void printPermutations(String str) {
        printPermutationsHelper(str, 0, str.length() - 1);
    }
    private static void printPermutationsHelper(String str, int left, int right) {
        if (left == right) {
            System.out.println(str);
        } else {
            for (int i = left; i <= right; i++) {
                str = swap(str, left, i);
                printPermutationsHelper(str, left + 1, right);
                str = swap(str, left, i); // backtrack
            }
        }
    }
    private static String swap(String str, int i, int j) {
        char[] charArray = str.toCharArray();
        char temp = charArray[i];
        charArray[i] = charArray[j];
        charArray[j] = temp;
        return new String(charArray);
    }
}
```

```
cal_percentage.java  LongestSubstringWithoutDu...  FibonacciSeries.java  StringPermutations.java X
11
12     System.out.println("Permutations of the string:");
13     printPermutations(input);
14
15     scanner.close();
16 }
17
18 public static void printPermutations(String str) {
19     printPermutationsHelper(str, 0, str.length() - 1);
20 }
21
22 private static void printPermutationsHelper(String str, int left, int right) {
23     if (left == right) {
24         System.out.println(str);
25     } else {
26         for (int i = left; i <= right; i++) {
27             str = swap(str, left, i);
28             printPermutationsHelper(str, left + 1, right);
29             str = swap(str, left, i); // backtrack
30         }
31     }
32 }
33
34 private static String swap(String str, int i, int j) {
35     char[] charArray = str.toCharArray();
36     char temp = charArray[i];
37     charArray[i] = charArray[j];
38     charArray[j] = temp;
39     return new String(charArray);
40 }
41 }

@ Javadoc Declaration Console X Coverage TestNG
<terminated> StringPermutations [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Dec 21, 2023, 9:50:06 PM)
Enter a string: cat
Permutations of the string:
cat
cta
act
atc
tac
tca
```

cal_percentage.javaLongestSubstringWithoutDu...FibonacciSeries.javaStringPermutations.java ×

```
1 package package3;
2
3 import java.util.Scanner;
4
5 public class StringPermutations {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8
9         System.out.print("Enter a string: ");
10        String input = scanner.nextLine();
11
12        System.out.println("Permutations of the string:");
13        printPermutations(input);
14
15        scanner.close();
16    }
17
18    public static void printPermutations(String str) {
19        printPermutationsHelper(str, 0, str.length() - 1);
20    }
21
22    private static void printPermutationsHelper(String str, int left, int right) {
23        if (left == right) {
24            System.out.println(str);
25        } else {
26            for (int i = left; i <= right; i++) {
27                str = swap(str, left, i);
28                printPermutationsHelper(str, left + 1, right);
29                str = swap(str, left, i); // backtrack
30            }
31        }
32    }
33}
```

@ Javadoc Declaration Console × Coverage TestNG

<terminated> StringPermutations [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Dec 21, 2023, 9:50:06 PM -

Enter a string: cat
Permutations of the string:
cat
cta
act
atc
tac
tca

Q4: Write a program to check if two strings are Anagrams?

Example:

- Input : Welcome
- Output : ceelmow

```
package package3;
import java.util.Arrays;
import java.util.Scanner;
public class AnagramChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the first string: ");
        String str1 = scanner.nextLine().toLowerCase();
        System.out.print("Enter the second string: ");
        String str2 = scanner.nextLine().toLowerCase();
        if (areAnagrams(str1, str2)) {
            System.out.println("The strings are Anagrams.");
        } else {
            System.out.println("The strings are not Anagrams.");
        }
        scanner.close();
    }
    public static boolean areAnagrams(String str1, String str2) {
        // Remove spaces and convert to lowercase
        str1 = str1.replaceAll("\\s", "");
        str2 = str2.replaceAll("\\s", "");
        // Check if lengths are equal
        if (str1.length() != str2.length()) {
            return false;
        }
        // Convert strings to char arrays and sort them
        char[] charArray1 = str1.toCharArray();
        char[] charArray2 = str2.toCharArray();
        Arrays.sort(charArray1);
        Arrays.sort(charArray2);
        // Compare sorted char arrays
        return Arrays.equals(charArray1, charArray2);
    }
}
```

cal_percentage.java LongestSubstringWithoutDu... FibonacciSeries.java StringPermutations.java AnagramChecker.java X

```
9
10     System.out.print("Enter the first string: ");
11     String str1 = scanner.nextLine().toLowerCase();
12
13     System.out.print("Enter the second string: ");
14     String str2 = scanner.nextLine().toLowerCase();
15
16     if (areAnagrams(str1, str2)) {
17         System.out.println("The strings are Anagrams.");
18     } else {
19         System.out.println("The strings are not Anagrams.");
20     }
21
22     scanner.close();
23 }
24
25 public static boolean areAnagrams(String str1, String str2) {
26     // Remove spaces and convert to lowercase
27     str1 = str1.replaceAll("\\s", "");
28     str2 = str2.replaceAll("\\s", "");
29
30     // Check if lengths are equal
31     if (str1.length() != str2.length()) {
32         return false;
33     }
34
35     // Convert strings to char arrays and sort them
36     char[] charArray1 = str1.toCharArray();
37     char[] charArray2 = str2.toCharArray();
38
39     Arrays.sort(charArray1);
40     Arrays.sort(charArray2);
41
42     // Compare sorted char arrays
43     return Arrays.equals(charArray1, charArray2);
44 }
45 }
```

@ Javadoc Declaration Console X Coverage TestNG

<terminated> AnagramChecker [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Dec 21, 2023, 9:51:23 PM – 9:51:48 PM) [pid: 15472]

Enter the first string: welcome
Enter the second string: elcomew
The strings are Anagrams.


```
cal_percentage.java  LongestSubstringWithoutDu...  FibonacciSeries.java  StringPermutations.java  AnagramChecker.java X
1 package package3;
2
3 import java.util.Arrays;
4 import java.util.Scanner;
5
6 public class AnagramChecker {
7     public static void main(String[] args) {
8         Scanner scanner = new Scanner(System.in);
9
10        System.out.print("Enter the first string: ");
11        String str1 = scanner.nextLine().toLowerCase();
12
13        System.out.print("Enter the second string: ");
14        String str2 = scanner.nextLine().toLowerCase();
15
16        if (areAnagrams(str1, str2)) {
17            System.out.println("The strings are Anagrams.");
18        } else {
19            System.out.println("The strings are not Anagrams.");
20        }
21
22        scanner.close();
23    }
24
25    public static boolean areAnagrams(String str1, String str2) {
26        // Remove spaces and convert to lowercase
27        str1 = str1.replaceAll("\\s", "");
28        str2 = str2.replaceAll("\\s", "");
29
30        // Check if lengths are equal
31        if (str1.length() != str2.length()) {
32            return false;
33        }
34
35        // Convert strings to char arrays and sort them
36        char[] charArray1 = str1.toCharArray();
37        char[] charArray2 = str2.toCharArray();
38
39        Arrays.sort(charArray1);
40        Arrays.sort(charArray2);
41
42        for (int i = 0; i < charArray1.length; i++) {
43            if (charArray1[i] != charArray2[i]) {
44                return false;
45            }
46        }
47
48        return true;
49    }
50}
```

@ Javadoc Declaration Console X Coverage TestNG

<terminated> AnagramChecker [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Dec 21, 2023, 9:51:23 PM – 9:51:48 PM) [pid: 15472]

Enter the first string: welcome

Enter the second string: elcomew

The strings are Anagrams.

Q5: Write a program to reverse a string.

Example:

- Input : Welcome
- Output : emoclew

```
package package3;
import java.util.Scanner;
public class StringReversal {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();
        String reversedString = reverseString(input);
        System.out.println("Reversed String: " + reversedString);
        scanner.close();
    }
    public static String reverseString(String str) {
        char[] charArray = str.toCharArray();
        int start = 0;
        int end = str.length() - 1;
        while (start < end) {
            // Swap characters at start and end indices
            char temp = charArray[start];
            charArray[start] = charArray[end];
            charArray[end] = temp;
            // Move indices towards the center
            start++;
            end--;
        }
        return new String(charArray);
    }
}
```

```
cal_percentage.java  LongestSubstringWithoutDu...  FibonacciSeries.java  StringPermutations.java  AnagramChecker.java  StringReversal.java x
9      System.out.print( "Enter a string: ");
10     String input = scanner.nextLine();
11
12     String reversedString = reverseString(input);
13
14     System.out.println("Reversed String: " + reversedString);
15
16     scanner.close();
17 }
18
19 public static String reverseString(String str) {
20     char[] charArray = str.toCharArray();
21
22     int start = 0;
23     int end = str.length() - 1;
24
25     while (start < end) {
26         // Swap characters at start and end indices
27         char temp = charArray[start];
28         charArray[start] = charArray[end];
29         charArray[end] = temp;
30
31         // Move indices towards the center
32         start++;
33         end--;
34     }
35
36     return new String(charArray);
37 }
38 }
<
@ Javadoc Declaration Console x Coverage TestNG
<terminated> StringReversal [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Dec 21, 2023, 9:48:33 PM - 9:48:40 PM) [pid: 9296]
Enter a string: welcome
Reversed String: emoclew
```

```
cal_percentage.java  LongestSubstringWithoutDu...  FibonacciSeries.java  StringPermutations.java  AnagramChecker.java  StringReversal.java x  StackUsingQueues.java
1 package package3;
2
3 import java.util.Scanner;
4
5 public class StringReversal {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8
9         System.out.print("Enter a string: ");
10        String input = scanner.nextLine();
11
12        String reversedString = reverseString(input);
13
14        System.out.println("Reversed String: " + reversedString);
15
16        scanner.close();
17    }
18
19    public static String reverseString(String str) {
20        char[] charArray = str.toCharArray();
21
22        int start = 0;
23        int end = str.length() - 1;
24
25        while (start < end) {
26            // Swap characters at start and end indices
27            char temp = charArray[start];
28            charArray[start] = charArray[end];
29            charArray[end] = temp;
30        }
31
32        return new String(charArray);
33    }
34 }
<
@ Javadoc Declaration Console x Coverage TestNG
<terminated> StringReversal [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (Dec 21, 2023, 9:48:33 PM - 9:48:40 PM) [pid: 9296]
Enter a string: welcome
Reversed String: emoclew
```

Q6:Implement stack using queue without using an array or linked list.

- Push
- Pop
- Top

```
package package3;
import java.util.LinkedList;
import java.util.Queue;
class StackUsingQueues {
    private Queue<Integer> mainQueue;
    private Queue<Integer> tempQueue;
    public StackUsingQueues() {
        mainQueue = new LinkedList<>();
        tempQueue = new LinkedList<>();
    }
    // Push element onto the stack
    public void push(int x) {
        // Move all elements from mainQueue to tempQueue
        while (!mainQueue.isEmpty()) {
            tempQueue.offer(mainQueue.poll());
        }
        // Add the new element to mainQueue
        mainQueue.offer(x);
        // Move elements back from tempQueue to mainQueue
        while (!tempQueue.isEmpty()) {
            mainQueue.offer(tempQueue.poll());
        }
    }
    // Removes the element on the top of the stack
    public void pop() {
        if (!mainQueue.isEmpty()) {
            mainQueue.poll();
        } else {
            System.out.println("Stack is empty. Cannot perform pop operation.");
        }
    }
    // Returns the element on the top of the stack
    public int top() {
        if (!mainQueue.isEmpty()) {
            return mainQueue.peek();
        } else {
            System.out.println("Stack is empty. Cannot perform top operation.");
            return -1; // indicating an empty stack
        }
    }
    // Returns true if the stack is empty
    public boolean isEmpty() {
        return mainQueue.isEmpty();
    }
}
```

```
public static void main(String[] args) {  
    StackUsingQueues stack = new StackUsingQueues();  
    stack.push(1);  
    stack.push(2);  
    stack.push(3);  
    System.out.println("Top element: " + stack.top()); // Output: 3  
    stack.pop();  
    System.out.println("Top element after pop: " + stack.top()); // Output: 2  
    stack.pop();  
    stack.pop();  
    System.out.println("Is the stack empty? " + stack.isEmpty()); // Output: true  
}  
}
```