

```
1 package com.oddeven;
2
3 import java.util.Arrays;
4
5 public class Anagram {
6
7     static void isAnagram(String str1, String str2) {
8         String s1 = str1.replaceAll("\\s", "");
9         String s2 = str2.replaceAll("\\s", "");
10        boolean status = true;
11        if (s1.length() != s2.length()) {
12            status = false;
13        } else {
14            char[] ArrayS1 = s1.toLowerCase().toCharArray();
15            char[] ArrayS2 = s2.toLowerCase().toCharArray();
16            Arrays.sort(ArrayS1);
17            Arrays.sort(ArrayS2);
18            status = Arrays.equals(ArrayS1, ArrayS2);
19        }
20        if (status) {
21            System.out.println(s1 + " and " + s2 + " are anagrams");
22        } else {
23            System.out.println(s1 + " and " + s2 + " are not anagrams");
24        }
25    }
26
27
28
29    public static void main(String[] args) {
30        isAnagram("Keep", "Peek");
31    }
32
33
```

```
1 package com.oddeven;
2
3 import java.util.Scanner;
4
5 public class ArmStrong {
6
7     public static void main(String[] args) {
8         int n =153;
9         int temp = n;
10        int r, sum = 0;
11
12        while (n > 0) {
13            r = n % 10;
14            n = n / 10;
15            sum = sum + r * r * r;
16        }
17        if (temp == sum) {
18            System.out.println("This is armstrong number");
19        } else {
20            System.out.println("Not armstrong number ");
21        }
22    }
23
24 }
25
26
```

```
1 package com.oddeven;
2
3 public class EvenOdd {
4
5     public static void main(String[] args) {
6         CountObj ob = new CountObj();
7
8         int no = 23;
9         if (no % 2 == 0)
10
11         {
12             System.out.println("This is even No");
13
14         }
15
16         else {
17
18             System.out.println("This is odd No");
19
20         }
21     }
22 }
23
24
```

```
1 package com.oddeven;
2
3 public class PyramidPattern {
4
5     public static void printTriangle(int n) {
6         for (int i = 0; i < n; i++) {
7             for (int j = n - i; j > 1; j--) {
8                 System.out.print(" ");
9             }
10            for (int j = 0; j <= i; j++) {
11                System.out.print("* ");
12            }
13            System.out.println();
14        }
15    }
16
17    public static void main(String args[]) {
18        int n = 10;
19        printTriangle(n);
20    }
21 }
22
```

<terminated> PyramidPattern [Java Application] /Users/shubham/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86\_64\_18.0.1.v2

```

  *
 * *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
* * * * * * * *
* * * * * * * * *
* * * * * * * * * *
```

```
1 package com.oddeven;
2
3 public class Factorial {
4
5     public static void main(String[] args) {
6         printfact(9);
7     }
8
9
10    static void printfact(int n) {
11        int fact = 1;
12        for (int i = 1; i <= n; i++) {
13            fact = fact * i;
14            System.out.println(fact);
15        }
16    }
17 }
18
```



```
1 package com.oddeven;
2
3
4 public class Fibonacci {
5
6     public static void Fibonacci(int N) {
7         int num1 = 0, num2 = 1, counter = 0;
8
9         while (counter < N ) {
10
11             int num3 = num2 + num1;
12             num1 = num2;
13             num2 = num3;
14             counter++;
15             System.out.println(num1);
16         }
17
18     }
19
20     public static void main(String[] args) {
21
22         Fibonacci(15);|
23
24     }
25
26 }
27
```



Immutableclass.java ×

```
1 package com.oddeven;
2
3
4 final class Immutableclass {
5
6     private String name ;
7
8     private String getName() {
9         return name;
10    }
11
12 }
13
```

```
1 package com.oddeven;
2
3 public class LeapYr {
4     public static void main(String[] args) {
5         int yr = 2000;
6
7         if (((yr % 4 == 0) && (yr % 100 != 0) || (yr % 400 == 0))) {
8             System.out.println("Leap");
9
10        }
11
12        else {
13            System.out.println("Not leap");
14
15        }
16    }
17 }
18
19
```



```
1 package com.oddeven;
2
3 import java.util.ArrayList;
4
5 public class List2Array {
6     public static void main(String args[]) {
7         // Instantiating and initializing ArrayList
8         ArrayList<String> cities = new ArrayList<>();
9         cities.add("Boston");
10        cities.add("Dallas");
11        cities.add("San jose");
12        cities.add("Chicago");
13
14        // ArrayList to String array conversion using toArray()
15        String citinames[]=cities.toArray(new String[cities.size()]);
16
17        // Printing elements using for-each loop
18        for(String str : citinames) {
19            System.out.println(str);
20        }
21    }
22 }
```

```
1 package com.oddeven;
2
3 public class PalindromeNumber {
4
5     public static void main(String[] args) {
6         int number = 12321, reverse = 0;
7         int temp = number;
8         while (number != 0) {
9             int remainder = number % 10;
10            number = number / 10;
11            reverse = reverse * 10 + remainder;
12        }
13
14
15        if(temp==reverse) {
16            System.out.println("palindrome number "); }
17        else {
18            System.out.println("not palindrome");
19        }
20    }
21 }
22
23
24
25
```

```
1 package com.oddeven;
2
3 public class ReverseNumber {
4
5     public static void main(String[] args) {
6         int number = 123, reverse = 0;
7
8         while (number != 0) {
9             int remainder = number % 10;
10            number = number / 10;
11            reverse = reverse * 10 + remainder;
12
13        }
14
15        System.out.println("Reverse Number is " + reverse);
16    }
17 }
18
```

```
1 package com.oddeven;
2 public class Primenumber {
3
4     public static boolean isPrime(int n) {
5
6         if (n < 2) {
7             return false;
8         }
9
10        for (int i = 2; i <= n / 2; i++) {
11            if (n % i == 0) {
12                return false;
13            }
14        }
15        return true;
16    }
17
18    public static void main(String[] args) {
19
20        int n = 4;
21        if (isPrime(n)) {
22            System.out.println(n + " is prime ");
23        } else {
24            System.out.println(n + " not prime");
25        }
26    }
27 }
28
```



```
1 package com.oddeven;
2
3 public class ReverseString {
4     public static void main(String[] args) {
5
6         String str = "Shubham", str2 = "";
7         char ch;
8
9         System.out.println("This is original word>>" + str);
10
11         for (int i = 0; i < str.length(); i++) {
12             ch = str.charAt(i);
13             str2 = ch + str2;
14         }
15         System.out.println("This is Reversed word>>" + str2);
16         System.out.println("Length of word>>" + str2.length());
17     }
18 }
19
```



```
1 package com.shubham;
2
3 import java.util.Scanner;
4
5 public class StringCharacterCount2 {
6     public static void main(String args[]) {
7         String str;
8         int i, length, counter[] = new int[256];
9
10        Scanner scanner = new Scanner(System.in);
11        System.out.println("Enter a String");
12        str = scanner.nextLine();
13
14        length = str.length();
15
16        // Count frequency of every character and store
17        // it in counter array
18        for (i = 0; i < length; i++) {
19            counter[(int) str.charAt(i)]++;
20        }
21        // Print Frequency of characters
22        for (i = 0; i < 256; i++) {
23            if (counter[i] != 0) {
24                System.out.println((char) i + " --> " + counter[i]);
25            }
26        }
27    }
28 }
```

<terminated> StringCharacterCount [Java Application] /Users/shubham/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86\_64\_18

Enter the string>>

Shubhammmmm

S occurs 1 time(s)

h occurs 2 time(s)

u occurs 1 time(s)

b occurs 1 time(s)

a occurs 1 time(s)

m occurs 2 time(s)

m occurs 2 time(s)

m occurs 1 time(s)



StringReverse.java X

```
1 package com.oddeven;
2
3 public class StringReverse {
4
5     public static void main(String[] args) {
6         StringBuffer sb = new StringBuffer("12345");
7         sb.reverse();
8         System.out.println("reversed >>>" + sb);
9
10    }
11 }
12
```

```
1 package com.oddeven;
2
3 public class VowelCount {
4
5     public static void main(String[] args) {
6
7         int vCount = 0;
8
9         String str = "This is a really simple sentence";
10
11         str = str.toLowerCase();
12
13         for (int i = 0; i < str.length(); i++) {
14
15             if (str.charAt(i) == 'a' || str.charAt(i) == 'e'
16                 || str.charAt(i) == 'i' || str.charAt(i) == 'o'
17                 || str.charAt(i) == 'u') {
18
19                 vCount++;
20             }
21         }
22         System.out.println("Number of vowels: " + vCount);
23     }
24
25 }
26
```

```
1 package com.oddeven;
2
3 public class VowelUpperCase {
4
5     static void conVowUpp(char[] str) {
6         int N = str.length;
7
8         for (int i = 0; i < N; i++) {
9             if (str[i] == 'a' || str[i] == 'e' || str[i] == 'i'
10                || str[i] == 'o' || str[i] == 'u')
11             {
12                 char c = Character.toUpperCase(str[i]);
13                 str[i] = c;
14             }
15         }
16         for (char c : str)
17             System.out.print(c);
18     }
19
20     // Driver Code
21     public static void main(String[] args) {
22         String str = "eutopia";
23         conVowUpp(str.toCharArray());
24     }
25 }
26
```



```
1 package com.oddeven;
2
3 public class SecondLargestInArray {
4
5     public static int getSecondLargest(int[] a, int total) {
6         int temp;
7         for (int i = 0; i < total; i++) {
8             for (int j = i + 1; j < total; j++) {
9                 if (a[i] > a[j]) {
10                     temp = a[i];
11                     a[i] = a[j];
12                     a[j] = temp;
13                 }
14             }
15         }
16         return a[total - 2];
17     }
18
19     public static void main(String args[]) {
20         int a[] = { 1, 2, 5, 6, 3, 2 };
21
22         System.out.println("Second Largest: " + getSecondLargest(a, 6));
23     }
24 }
25
26
```



```
1 package com.oddeven;
2
3 public class DuplicateElementsArray {
4
5     public static void main(String[] args) {
6
7         int[] arr = new int[] { 1, 2, 3, 4, 2, 7, 8, 8, 3 };
8
9         System.out.println("Duplicate elements in given array: ");
10
11
12         for (int i = 0; i < arr.length; i++) {
13             for (int j = i + 1; j < arr.length; j++) {
14                 if (arr[i] == arr[j])
15                     System.out.println(arr[j]);
16             }
17         }
18     }
19 }
20
21 |
```

<terminated> DuplicateElementsArray [Java Application] /Users/shubham/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86\_64.

Duplicate elements in given array:

2  
3  
8

```
1 package com.oddeven;
2
3 public class SmallestElementArray {
4
5     public static void main(String[] args) {
6
7         int[] arr = new int[] { 25, 11, 7, 75, 56 };
8
9         int min = arr[0];
10
11         for (int i = 0; i < arr.length; i++) {
12             if (arr[i] < min)
13                 min = arr[i];
14         }
15         System.out.println("Smallest element present in given array: " + min);
16     }
17
18 }
19 |
```

```
1 package com.oddeven;
2
3 public class LargestElementArray {
4
5     public static void main(String[] args) {
6
7
8         int [] arr = new int [] {25, 11, 7, 75, 56};
9
10        int max = arr[0];
11
12        for (int i = 0; i < arr.length; i++) {
13            if(arr[i] > max)
14                max = arr[i];
15        }
16
17        System.out.println("Largest element present in given array: " + max);
18    }
19 }
20
```

Largest element present in given array: 75

```
1 package com.oddeven;
2
3 public class ReplaceSpaceInString {
4
5
6     public static void main(String[] args) {
7         String string = "Once in a blue moon";
8         char ch = '-';
9
10        string = string.replace(' ', ch);
11
12        System.out.println("String after replacing spaces with given character");
13        System.out.println(string);
14    }
15 }
16
```

<terminated> ReplaceSpaceInString [Java Application] /Users/shubham/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86\_64\_18.0.

String after replacing spaces with given character:  
Once-in-a-blue-moon



```
6 ▶ public class Demo
7 {
8 ▶     public static void main(String[] args)
9     {
10         List<Integer> nums = Arrays.asList(4,5,7,8,9);
11
12         Stream<Integer> data = nums.stream();
13         long count = data.count();
14         System.out.println(count);
15 ⚡ data.forEach(n -> System.out.println(n));
16
17     }
18 }
19
```



```
1 package com.test;
2
3
4 interface Drawable{
5     public void draw();
6 }
7
8 public class demo{
9     public static void main(String[] args) {
10         String x ="This is lambda";
11
12         //with lambda
13         Drawable d2=()->System.out.println("Drawing "+ x );
14
15         d2.draw();
16     }
17 }
18 |
```

demo.java Gender.java test.java X

```
1 package com.test;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5 import java.util.List;
6 import java.util.stream.Collectors;
7 import java.util.stream.Collectors;
8
9 public class test {
10
11     public static void main(String[] args) {
12
13         ArrayList<demo> al = new ArrayList<>();
14
15         al.add(new demo(101,"Shubham",12000,Gender.Male));
16         al.add(new demo(102,"Shital",21000,Gender.Female));
17         al.add(new demo(103,"Shyam",8000,Gender.Male));
18         al.add(new demo(104,"Rahul",15000,Gender.Male));
19
20
21         // List<demo> list=al.stream().filter(e->e.getSalary()>=15000).collect(Collectors.toList());
22         //
23         // System.out.println(list);
24         //
25
26
27
28         Iterator itr=al.iterator();
29
30
31         while(itr.hasNext()){
32             demo emp = (demo)itr.next();
33             if(emp.getSalary()>10000) {
34                 System.out.println(emp.getSalary());
35             }
36         }
37
38     }
39 }
40
```

Problems Javadoc Declaration Console X

<terminated> test [Java Application] /Users/shubham/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86\_64\_18.0.1.v20220515-1614/jre/bin/java (31-Oct-2

12000  
21000  
15000