

SAVEETHA SCHOOL OF ENGINEERING
SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
CSA0987 –JAVA PROGRAMMING

NAME:K.Ganesh kumar

REG.NO:192365016

ASSIGNMENT:1

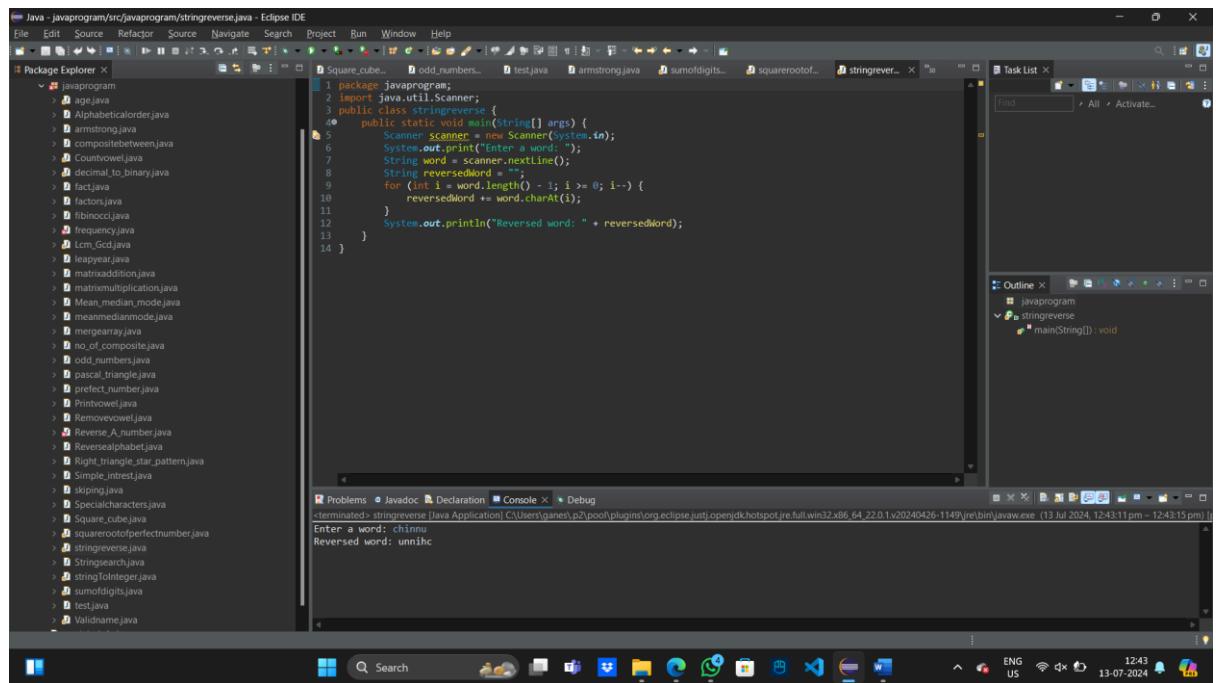
1. Write a program to reverse a word using loop? (Not to use inbuilt functions)

Sample Input:

String: TEMPLE

Sample Output:

Reverse String: ELPMET



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a large list of Java files under the package `javaprogram`, including `Square_cube.java`, `odd_numbers.java`, `test.java`, `armstrong.java`, `sumofdigits.java`, `squarerootof.java`, and `stringreverse.java`.
- Code Editor:** Displays the `stringreverse.java` file with the following code:

```
1 package javaprogram;
2 import java.util.Scanner;
3 public class stringreverse {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner("Enter a word: ");
6         System.out.print("Enter a word: ");
7         String word = scanner.nextLine();
8         String reversedWord = "";
9         for (int i = word.length() - 1; i >= 0; i--) {
10             reversedWord += word.charAt(i);
11         }
12         System.out.println("Reversed word: " + reversedWord);
13     }
14 }
```
- Console:** Shows the output of the program running in the terminal window:

```
Enter a word: chinmu
Reversed word: ummihc
```

2. Write a program to convert the given string to integer?

Sample Input:

String: 1234

Sample Output:

Output String: 1234

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a package named "javaprogram" containing numerous Java files such as age.java, Alphabeticalorder.java, armstrong.java, compositebetween.java, Countownto.java, decimal_to_binary.java, fact.java, factors.java, fibonacci.java, frequency.java, Lcm_Gcd.java, leapyear.java, matrixaddition.java, matrixmultiplication.java, Mean_median_mode.java, meandomediammode.java, mergearray.java, no_of_composite.java, odd_numbers.java, pascal_triangle.java, prefect_number.java, Printnow.java, Removewowel.java, Reverse_A_number.java, Reversealphabet.java, Right_triangle_star_pattern.java, Simple_interest.java, Specialcharacters.java, Square_cube.java, squarerootofperfectnumber.java, Stringreverse.java, Stringsearch.java, stringToInteger.java, sumofdigits.java, test.java, and Validname.java.
- Code Editor:** Displays the "stringToInteger.java" file with the following code:

```
1 package Javaprogram;
2
3 public class stringToInteger {
4     public static void main(String args[]) {
5         int decimalExample = Integer.parseInt("20");
6         int signedPositiveExample = Integer.parseInt("+20");
7         System.out.println(decimalExample);
8     }
9 }
10
11
```
- Console:** Shows the output of the program: "20".
- Task List:** Shows a single task: "Find All Activate...".
- Outline:** Shows the class structure: "javaprogram", "stringToInteger", and "main(String[])".

3. Write a program to check the entered user name is valid or not. Get both the inputs from the user.

```

1 package javaprogram;
2
3 import java.util.Scanner;
4
5 public class Validname {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8         String username = scanner.nextLine();
9         if (isValidUsername(username)) {
10             System.out.println("Valid username.");
11         } else {
12             System.out.println("Invalid username.");
13         }
14     }
15     public static boolean isValidUsername(String username) {
16         // Check length
17         if (username.length() < 5 || username.length() > 15) {
18             return false;
19         }
20         if (!username.matches("[a-zA-Z0-9]+")) {
21             return false;
22         }
23         if (Character.isDigit(username.charAt(0))) {
24             return false;
25         }
26         return true;
27     }
28 }

```

The console output shows the program running and accepting the input "konda". It then prints "Valid username.".

4. Write a program that would sort a list of names in alphabetical order Ascending or Descending, choice get from the user?

Sample Input:

Banana

Carrot

Radish

Apple

Jack

Order(A/D) : A

Sample Output:

Apple

Banana

Carrot

Jack

Radish

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Java - javaprogram/src/javaprogram/Alphabeticalorder.java - Eclipse IDE
- Package Explorer:** Shows multiple Java files under the package javaprogram.
- Code Editor:** Displays the following Java code for `Alphabeticalorder`:

```
1 package javaprogram;
2
3 public class Alphabeticalorder {
4     public static void main(String[] args) {
5         String[] names = {"konda", "siva", "rama", "jeshwanth"};
6         int m;
7         String temp;
8         for(int i=0;i<n;i++) {
9             for(int j=0;j<n;j++) {
10                 if(names[i].compareTo(names[j])>0) {
11                     temp=names[i];
12                     names[i]=names[j];
13                     names[j]=temp;
14                 }
15             }
16         }
17         for(int i=0;i<n;i++) {
18             System.out.println(names[i]);
19         }
20     }
21 }
```

- Console:** Shows the output of the program: siva, rama, konda, jeshwanth.
- Bottom Status Bar:** Shows the date and time: 13-07-2024, 12:48:41 pm – 12:48:42 pm.

5. Write a program to print the special characters separately and print number of Special characters in the line?

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Java - javaprogram/src/javaprogram/Specialcharacters.java - Eclipse IDE
- Package Explorer:** Shows multiple Java files under the package javaprogram.
- Code Editor:** Displays the following Java code for `Specialcharacters`:

```
1 package javaprogram;
2
3 public class Specialcharacters {
4     public static void main(String[] args) {
5         System.out.print("Please Enter Alpha Numeric Special String: ");
6         String str = "Chinu@206";
7         int alpha = 0, digit = 0, spl = 0;
8         for (int i = 0; i < str.length(); i++) {
9             char ch = str.charAt(i);
10             if (Character.isLetter(ch)) {
11                 alpha++;
12             } else if (Character.isDigit(ch)) {
13                 digit++;
14             } else {
15                 spl++;
16             }
17         }
18         System.out.println("Number of Alphabet Characters: " + alpha);
19         System.out.println("Number of Digit Characters: " + digit);
20         System.out.println("Number of Special Characters: " + spl);
21     }
22 }
```

- Console:** Shows the output of the program: Please Enter Alpha Numeric Special String: Number of Alphabet Characters: 6 Number of Digit Characters: 4 Number of Special Characters: 1
- Bottom Status Bar:** Shows the date and time: 13-07-2024, 12:49:29 pm – 12:49:30 pm.

6. Write a program to print the number of vowels in the given statement?

Sample Input:

Saveetha School of Engineering

Sample Output:

Number of vowels = 12

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Java - javaprogram/src/javaprogram/Countvowel.java - Eclipse IDE
- Menu Bar:** File, Edit, Source, Refactor, Source, Navigate, Project, Run, Window, Help
- Toolbar:** Standard Eclipse toolbar icons.
- Package Explorer:** Shows a large list of Java files under the package `javaprogram`, including `Countvowel.java`.
- Editor:** Displays the code for `Countvowel.java`. The code initializes a string `str` to "Chinnu", converts it to uppercase, and then iterates through each character. If the character is 'a' or 'e' or 'i' or 'o' or 'u', it increments a counter. Finally, it prints the total count.
- Outline View:** Shows the class `Countvowel` and its main method.
- Task List:** Empty.
- Console:** Shows the output of the program: "Total no of vowels in string are: 12".
- Bottom Status Bar:** Shows system information like date, time, and battery level.

7. Write a program to print consonants and vowels separately in the given word

Sample Input:

Given Word: Engineering

Sample Output:

Consonants: n g n r n g

Vowels: e i e ei

```

Java - javaprogram/src/javaprogram/Printvowel.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer X
Stringtolint... Validname.java Alphabetical... Printvowel.java Specialchara... stringrever...
String str = "Konda";
str = str.toLowerCase();
System.out.println("vowels:");
for (int i = 0; i < str.length(); i++)
{
    if (str.charAt(i) == 'a' || str.charAt(i) == 'e'
    || str.charAt(i) == 'i'
    || str.charAt(i) == 'o'
    || str.charAt(i) == 'u')
    {
        System.out.println(str.charAt(i));
    }
}
System.out.println("consonent:");
for (int i = 0; i < str.length(); i++)
{
    if (str.charAt(i) == 'a' || str.charAt(i) == 'e'
    || str.charAt(i) == 'i'
    || str.charAt(i) == 'o'
    || str.charAt(i) == 'u')
    {
        continue;
    }
    else
    {
        System.out.println(str.charAt(i));
    }
}
}

```

Problems Javadoc Declaration Console Debug
terminated: Printvowel [Java Application] C:\Users\ganes\Downloads\javaprogram\src\javaprogram\Printvowel.java
vowels:
o
a
consonent:
k
n
d

8. Write a program that finds whether a given character is present in a string or not. In case it is present it prints the index at which it is present. Do not use built-in find functions to search the character.

Sample Input:

Enter the string: I am a programmer

Enter the character to be searched: p

Sample Output:

P is found in string at index: 8

Note: Check for non available Character in the given statement as Hidden Test case.

```

1 package javaprogram;
2
3 public class Stringsearch{
4     public static void main(String[] args) {
5         String inputString = "Hello, world!";
6         char targetChar = 'o';
7         int index = findCharacterIndex(inputString, targetChar);
8         if (index != -1) {
9             System.out.println("Character '" + targetChar + "' found at index: " + index);
10        } else {
11            System.out.println("Character '" + targetChar + "' not found in the string.");
12        }
13    }
14    public static int findCharacterIndex(String str, char target) {
15        char[] charArray = str.toCharArray();
16        for (int i = 0; i < charArray.length; i++) {
17            if (charArray[i] == target) {
18                return i;
19            }
20        }
21        return -1;
22    }
23 }
24
25

```

Character 'o' found at index: 4

9. Write a program to arrange the letters of the word alphabetically in reverse order

Sample Input:

Enter the word: MOSQUE

Sample Output:

Alphabetical Order: U S Q O M E

Test Case:

1. HYPOTHECATION
2. MATRICULATION
3. MANIPULATION

```

Java - javaprogram/src/javaprogram/alphabeticalreverse.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer X
Alphabetical... Printvowel.java Countvowel.java Specialchara... Stringsearc... alphabetical... X Task List X
Find All Activate...
Outline X
javaprogram alphabeticalreverse
main(String[])

```

```

1 package javaprogram;
2 import java.util.Arrays;
3 import java.util.Scanner;
4 public class alphabeticalreverse {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         System.out.print("Enter a word: ");
8         String word = scanner.nextLine();
9         char[] charArray = word.toCharArray();
10        Arrays.sort(charArray);
11        StringBuilder reversedWord = new StringBuilder(new String(charArray));
12        reversedWord.reverse();
13        System.out.println("Reversed word in alphabetical order: " + reversedWord.toString());
14    }
15 }

```

Problems Javadoc Declaration Console Debug
<terminated> alphabeticalreverse [Java Application] C:\Users\panes\p2\pool\plugins\org.eclipse.jdt.core\full\win32\x86_64_22.0.1.v20240426-1149\releas\bin\javaw.exe (13 Jul 2024, 12:56:52 pm – 12:56:52 pm)
Enter a word: chinna
Reversed word in alphabetical order: unnihc

10. Write a program that accepts a string from user and displays the same string after removing vowels from it.

Sample Input & Output:

Enter a string: we can play the game

The string without vowels is: w cn pl thgm

Arrays:

```

Java - javaprogram/src/javaprogram/Removevowel.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer X
Removevowel.java Countvowel.java Stringsearch... Removevowel.java alphabetical... 12
1 package javaprogram;
2
3 public class Removevowel {
4
5     String s = "chinnu";
6
7     for (int i = 0; i < s.length(); i++) {
8         if (s.charAt(i) == 'a' || s.charAt(i) == 'e'
9             || s.charAt(i) == 'i' || s.charAt(i) == 'o'
10            || s.charAt(i) == 'u' || s.charAt(i) == 'A'
11            || s.charAt(i) == 'E' || s.charAt(i) == 'I'
12            || s.charAt(i) == 'O'
13            || s.charAt(i) == 'U') {
14             continue;
15         } else {
16             System.out.print(s.charAt(i));
17         }
18     }
19 }
20 }
21

```

Console tab output:

```

<terminated> Removevowel[Java Application] C:\Users\ganes\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64_22.0.1.v20240426-1149\jre\bin\javaw.exe (13 Jul 2024, 12:58:06 pm - 12:58:06 pm)
chnn

```

ASSIGNMENT:2

1. Write a program for matrix multiplication?

Sample Input:

Mat1 = 1 2

 5 3

Mat2 = 2 3

 4 1

Sample Output:

Mat Sum = 10 5

 22 18

```

Java - javaprogram/src/javaprogram/matrixmultiplication.java - Eclipse IDE
File Edit Source Refactor Source Navigate Project Run Window Help
Package Explorer X
Alphabetical... Countvoweljava Stringsearch... Removevowl... matrixmultiplication.java alphabetical... Task List X
Find All Activate...
1 package javaprogram;
2
3 public class matrixmultiplication {
4     public static void main(String[] args) {
5         int a1[][] = {{1,2},{2,1}};
6         int a2[][] = {{3,4},{4,3}};
7         int res[][]=new int[2][2];
8         for (int i=0;i<2;i++) {
9             for(int j=0;j<2;j++) {
10                 for(int k=0;k<2;k++) {
11                     res[i][j]+=a1[i][k]*a2[k][j];
12                 }
13             }
14         }
15         for(int[] row:res) {
16             for(int val:row) {
17                 System.out.print(val+" ");
18             }
19             System.out.println();
20         }
21     }
22 }

```

Console Output:

```

Problems Declaration Console Debug
<terminated> matrixmultiplication [Java Application] C:\Users\ganes\Pooh\plugins\org.eclipse.jdt.core\22.0.1.v20240426-1149\jre\bin\javaw.exe (13 Jul 2024, 12:59:19 pm - 12:59:21)
11 10
10 11

```

2. Write a program for matrix addition?

Sample Input:

Mat1 = 1 2

 5 3

Mat2 = 2 3

 4 1

Sample Output:

Mat Sum = 3 5

 9 4

```

Java - javaprogram/src/javaprogram/matrixaddition.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer X alphabetical... Countvowel... Removevowel... matrixmultip... matrixaddit... alphabetical... Task List X
1 package javaprogram;
2
3 public class matrixaddition {
4     public static void main(String[] args) {
5         int[][] mat1 = {{1, 2}, {3, 4}};
6         int[][] mat2 = {{2, 3}, {4, 1}};
7         int[][] result = new int[2][2];
8
9         for (int i = 0; i < 2; i++) {
10            for (int j = 0; j < 2; j++) {
11                result[i][j] = mat1[i][j] + mat2[i][j];
12            }
13        }
14        System.out.println("Matrix Addition Result:");
15        for (int[] row : result) {
16            for (int val : row) {
17                System.out.print(val + " ");
18            }
19        }
20        System.out.println();
21    }
22}
23

```

Console Output:

```

Matrix Addition Result:
3 5
9 4

```

3. Write a program for Merge two sorted arrays using Array list

Input: arr1[] = { 1, 3, 4, 5}, arr2[] = {2, 4, 6, 8}

Output: arr3[] = {1, 2, 3, 4, 4, 5, 6, 8}

```

Java - javaprogram/src/javaprogram/mergearray.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer X alphabetical... Removevowel... matrixmultip... matrixaddit... mergearray.java alphabetical... Task List X
1 package javaprogram;
2
3 import java.util.ArrayList;
4 public class mergearray {
5     public static void main(String[] args) {
6         int[] arr1 = {1, 3, 4, 5};
7         int[] arr2 = {2, 4, 6, 8};
8         ArrayList<Integer> mergedList = new ArrayList<>();
9         for (int val : arr1) {
10            mergedList.add(val);
11        }
12        for (int val : arr2) {
13            mergedList.add(val);
14        }
15        Collections.sort(mergedList);
16        System.out.println("Merged and Sorted Array:");
17        System.out.println(mergedList);
18    }
19 }
20

```

Console Output:

```

Merged and Sorted Array:
[1, 2, 3, 4, 4, 5, 6, 8]

```

4. Find the Mean, Median, Mode of the array of numbers?

Sample Input::

Array of elements = {16, 18, 27, 16, 23, 21, 19}

Sample Output: Mean = 20

Median = 19

Mode = 16

Test cases:

1. Array of elements = {26, 28, 37, 26, 33, 31, 29}

2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}

3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}

4. Array of elements = {200, 180, 180, 270, 160, 270, 270, 190, 200}

5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100, 100}

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows various Java files including `age.java`, `Alphabeticalorder.java`, `alphabeticalreverse.java`, `armstrong.java`, `compositebetween.java`, `Countvowel.java`, `decimal_to_binary.java`, `fact.java`, `factors.java`, `fibonacci.java`, `frequency.java`, `Lcm_Gcd.java`, `leapyear.java`, `matraddition.java`, `matrmultiplication.java`, `Mean_median_mode.java`, `meanmedianmode.java`, `mergearray.java`, `no_of_composite.java`, `odd_numbers.java`, `pascal_triangle.java`, `perfect_number.java`, `Printvowel.java`, `Removevowel.java`, `Reverse_A_number.java`, `Reversealphabct.java`, `Right_triangle_star_pattern.java`, `Simple_interest.java`, `skipping.java`, `Specialcharactersjava`, `Square_cube.java`, `squarerootofperfectnumber.java`, `stringreverse.java`, `Stringsearch.java`, `stringtointeger.java`, `sumofdigits.java`, `test.java`, and `Validname.java`.
- Code Editor:** Displays the `Mean_median_mode.java` file with the following code:

```
1 package javaprogram;
2 import java.util.*;
3 public class Mean_median_mode {
4     public static void main(String[] args) {
5         int[] array = { 1, 2, 3, 4, 5, 5, 6, 6, 6, 7 };
6         double mean = calculateMean(array);
7         System.out.println("Mean: " + mean);
8         double median = calculateMedian(array);
9         System.out.println("Median: " + median);
10        int mode = calculateMode(array);
11        System.out.println("Mode: " + mode);
12    }
13    public static double calculateMean(int[] array) {
14        int sum = 0;
15        for (int num : array) {
16            sum += num;
17        }
18        return (double) sum / array.length;
19    }
20    public static double calculateMedian(int[] array) {
21        Arrays.sort(array);
22        int middle = array.length / 2;
23        if (array.length % 2 == 0) {
24            double median = (array[middle - 1] + array[middle]) / 2.0;
25            return median;
26        } else {
27            return array[middle];
28        }
29    }
30    public static int calculateMode(int[] array) {
31        Map<Integer, Integer> frequencyMap = new HashMap<>();
32        for (int num : array) {
33            frequencyMap.put(num, frequencyMap.getOrDefault(num, 0) + 1);
34        }
35        int maxFrequency = 0;
36        int mode = 0;
37        for (Map.Entry<Integer, Integer> entry : frequencyMap.entrySet()) {
38            if (entry.getValue() > maxFrequency) {
39                maxFrequency = entry.getValue();
40                mode = entry.getKey();
41            }
42        }
43        return mode;
44    }
45 }
```

- Console:** Shows the output of the program:

```
Mean: 4.5
Median: 5.0
Mode: 6
```
- Task List:** Shows the tasks listed in the project.
- Outline:** Shows the class structure and methods.

5. Write a program to find the number of composite numbers in an array of elements

Sample Input::

Array of elements = {16, 18, 27, 16, 23, 21, 19}

Sample Output: Number of Composite Numbers = 5

Test cases:

1. Array of elements = {26, 28, 37, 26, 33, 31, 29}
2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}
3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}
4. Array of elements = {200, 180, 180, 270, 270, 270, 190, 200}
5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100}

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows various Java files in the `javaprogram` package.
- Code Editor:** Displays the `no_of_composite.java` file containing the following code:

```

1 package javaprogram;
2
3 public class no_of_composite {
4     public static void main(String[] args) {
5         int[] array = {16, 18, 27, 16, 23, 21, 19};
6         findCompositeNumbers(array);
7     }
8     public static void findCompositeNumbers(int[] array) {
9         int count = 0;
10        for (int num : array) {
11            if (isComposite(num)) {
12                System.out.println(num);
13            }
14        }
15    }
16    public static boolean isComposite(int num) {
17        if (num <= 1) return false;
18        for (int i = 2; i <= Math.sqrt(num); i++) {
19            if (num % i == 0) return true;
20        }
21        return false;
22    }
23 }

```

- Console:** Shows the output of the program execution:

```

16
18
27
16
21

```

6. Write a program to print Right Triangle Star Pattern

Sample Input:: n = 5

Output:

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

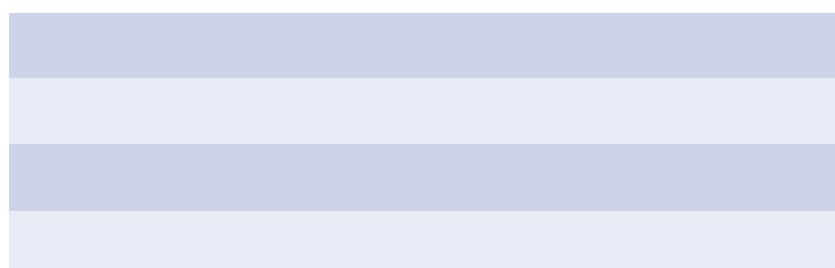
```
Java - javaprogram/src/javaprogram/Right_triangle_star_pattern.java - Eclipse IDE  
File Edit Source Refactor Source Navigate Project Run Window Help  
Package Explorer X matrixmult... matrixadditi... mergearray.java no_of_compo... Right_triang... Mean_median...  
matrixaddition.java  
Alphabeticorder.java  
alphabeticreverse.java  
armstrong.java  
compositebetween.java  
Countdown.java  
decimal_to_binary.java  
fact.java  
factors.java  
fibonacci.java  
frequency.java  
Lcm_Gcd.java  
leapyear.java  
matrixaddition.java  
matrixmultiplication.java  
Mean_median_mode.java  
meanmedianmode.java  
mergearray.java  
no_of_composite.java  
odd_numbers.java  
pascal_triangle.java  
prefect_number.java  
Printvowel.java  
Removewowel.java  
Reverse_A_number.java  
Reversealphabet.java  
Right_triangle_star_pattern.java  
Simple_intrest.java  
skipping.java  
Specialcharacters.java  
Square_cube.java  
squarerootofperfectnumber.java  
stringreverse.java  
Stringsearch.java  
stringToInteger.java  
sumofdigits.java  
test.java  
Validname.java  
javaprogram  
Right_triangle_star_pattern  
pattern(int)  
main(String[])
```

```
1 package javaprogram;  
2 public class Right_triangle_star_pattern {  
3     public static void pattern(int n) {  
4         for(int i=0;i<n; i++) {  
5             for(int b=0;b<i;b++) {  
6                 System.out.print(" ");  
7             }  
8             System.out.println();  
9         }  
10    }  
11    public static void main(String args[]) {  
12        int k=5;  
13        pattern(k);  
14    }  
15 }
```

Output in Console:
*
* *
* * *
* * * *
* * * * *

Bottom Status Bar: 11:42 : 285, ENG US, 13:05, 13-07-2024

7. Write a program to print the below pattern?



A screenshot of a Java code editor window. The title bar says "File Edit Selection View Go ...". The search bar says "java". The left sidebar shows an "EXPLORER" view with a "JAVA" folder containing "PascalTriangle.java". The main editor area contains the following Java code:

```
public class PascalTriangle {
    public static void main(String[] args) {
        int rows = 5;
        int[][] triangle = new int[rows][rows];
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j <= i; j++) {
                if (j == 0 || j == i) {
                    triangle[i][j] = 1;
                } else {
                    triangle[i][j] = triangle[i - 1][j - 1] + triangle[i - 1][j];
                }
            }
        }
        for (int i = 0; i < rows; i++) {
            for (int k = 0; k < rows - i - 1; k++) {
                System.out.print(" \t");
            }
            for (int j = 0; j <= i; j++) {
                System.out.print(triangle[i][j] + "\t\t");
            }
            System.out.println();
        }
    }
}
```

The status bar at the bottom shows "Ln 18, Col 1 Spaces: 4 UTF-8 CRLF {} Java".

8. Write a program to print rectangle symbol pattern.

Get the symbol as input from user

A screenshot of a Java code editor window. The title bar says "File Edit Selection View Go ...". The search bar says "java". The left sidebar shows an "EXPLORER" view with a "JAVA" folder containing "RectangleSymbolPattern.java". The main editor area contains the following Java code:

```
public class RectangleSymbolPattern {
    public static void main(String[] args) {
        String symbol = "*";
        int rows = 4;
        int columns = 6;
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= columns; j++) {
                System.out.print(symbol + " ");
            }
            System.out.println();
        }
    }
}
```

The status bar at the bottom shows "Ln 6, Col 1 Spaces: 4 UTF-8 CRLF {} Java".

9. Write a program to print the following pattern

Sample Input:

Enter the number to be printed: 1

Max Number of time printed: 3

1

11

111

11

1

The screenshot shows a Java code editor interface. The left sidebar has icons for Explorer, Java, Outline, Timeline, and Java Projects. The Java section is expanded, showing a file named 'NumberPattern.java'. The main editor area displays the following Java code:

```
public class NumberPattern {
    public static void main(String[] args) {
        int number = 1;
        int maxTimes = 3;
        for (int i = 1; i <= maxTimes; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(number);
            }
            System.out.println();
        }
        for (int i = maxTimes - 1; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print(number);
            }
            System.out.println();
        }
    }
}
```

The code uses nested loops to print a pattern of numbers. The first loop iterates from 1 to 3. Inside each iteration of the outer loop, a second loop prints the current value of 'number' (1, 2, or 3) 'i' times. After each iteration of the inner loop, a new line is printed. After the first three iterations, the pattern is printed again in reverse order from 3 down to 1.

10. Write a program to print the Inverted Full Pyramid pattern?

The screenshot shows the Microsoft Visual Studio Code interface. The left sidebar has a 'JAVA' section with a file named 'InvertedFullPyramid.java'. The main editor window displays the following Java code:

```
1 public class InvertedFullPyramid {
2     public static void main(String[] args) {
3         int rows = 5;
4         for (int i = rows; i >= 1; i--) {
5             for (int j = 1; j <= rows - i; j++) {
6                 System.out.print(" ");
7             }
8             for (int k = 1; k <= 2 * i - 1; k++) {
9                 System.out.print("#");
10            }
11            System.out.println();
12        }
13    }
14 }
```

The status bar at the bottom shows 'Ln 11, Col 1' and other system information like weather and date.

ASSIGNMENT:3

1. Write a program to print the following pattern

Sample Input:

Enter the Character to be printed: %

Max Number of time printed: 3

%

% %

% % %

The screenshot shows a Java development environment with the following details:

- File Menu:** File, Edit, Selection, View, Go, ...
- Toolbar:** Back, Forward, Search (with 'java' placeholder), Minimize, Maximize, Close.
- Code Editor:** CharacterPattern.java (Line 13: closing brace '}' is highlighted).

```
1 public class CharacterPattern {  
2     public static void main(String[] args) {  
3         char character = '%';  
4         int maxTimes = 3;  
5         for (int i = 1; i <= maxTimes; i++) {  
6             for (int j = 1; j <= i; j++) {  
7                 System.out.print(character + " ");  
8             }  
9         }  
10    }  
11 }  
12 }  
13 }
```
- Notifications:** A tooltip says "Build failed, do you want to continue? Source: Debugger for Java". Another tooltip says "Battery saver is on Consider plugging in your device."
- Bottom Bar:** ESP - FRA In 3 hours, Java: Ready, Search bar, Taskbar icons (Calculator, Mail, Google Chrome, File Explorer, Task View, Taskbar settings), Language: ENG IN, Date: 09-07-2024, Time: 21:45.

2. Write a program to print hollow square symbol pattern?

The screenshot shows a Java development environment with the following details:

- File Menu:** File, Edit, Selection, View, Go, ...
- Toolbar:** Back, Forward, Search (with 'java' placeholder), Minimize, Maximize, Close.
- Code Editor:** HollowSquarePattern.java (Line 10: cursor is at the start of the first '}' of the inner loop).

```
1 public class HollowSquarePattern {  
2     public static void main(String[] args) {  
3         char symbol = '*';  
4         int size = 5;  
5         for (int i = 1; i <= size; i++) {  
6             for (int j = 1; j <= size; j++) {  
7                 if (i == 1 || i == size || j == 1 || j == size) {  
8                     System.out.print(symbol + " ");  
9                 } else {  
10                     System.out.print(" ");  
11                 }  
12             }  
13         }  
14     }  
15 }  
16 }  
17 }
```
- Notifications:** A tooltip says "Build failed, do you want to continue? Source: Debugger for Java" with buttons Continue, Always Continue, Fix...".
- Bottom Bar:** Upcoming Earnings, Java: Ready, Search bar, Taskbar icons, Language: ENG IN, Date: 09-07-2024, Time: 21:47.

3. Write a program to print the below pattern

1

2 2

3 3 3

4 4 4 4

The screenshot shows the Microsoft Visual Studio Code interface. The left sidebar has 'EXPLORER' and 'JAVA' sections. The main area displays a Java file named 'NumberPattern.java' with the following code:

```
public class NumberPattern {
    public static void main(String[] args) {
        int rows = 4;
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(i + " ");
            }
            System.out.println();
        }
    }
}
```

The terminal at the bottom shows the output of running the program:

```
PS C:\Users\madet\OneDrive\Desktop\java> ^C
PS C:\Users\madet\OneDrive\Desktop\java>
PS C:\Users\madet\OneDrive\Desktop\java> c:: cd 'c:\Users\madet\OneDrive\Desktop\java'; & 'C:\Program Files\Java\jre-1.8\bin\java.exe' '-cp' 'C:\Users\madet\AppData\Roaming\Code\User\workspaceStorage\3d6ae272a81ab392ab799990cdee6162\redhat.java\jdt_ws\java_80cf0e31\bin' 'NumberPattern'
1
2 2
3 3 3
4 4 4 4
PS C:\Users\madet\OneDrive\Desktop\java>
```

The status bar at the bottom right shows the date and time: 09-07-2024 21:49.

4. Write a program to print the below pattern

1

4 9

16 25 36

49 64 81 100

The screenshot shows a Java code editor interface. The left sidebar has sections for EXPLORER, JAVA, OUTLINE, TIMELINE, and JAVA PROJECTS. The main area displays a Java file named SquarePattern.java with the following code:

```
public class SquarePattern {
    public static void main(String[] args) {
        int rows = 4;
        int start = 1;
        for (int i = 0; i < rows; i++) {
            int number = start + i * i;
            for (int j = 0; j <= i; j++) {
                System.out.print(number * number + " ");
                number += 2;
            }
            System.out.println();
        }
    }
}
```

The status bar at the bottom shows the current line (Ln 11, Col 35), spaces (Spaces: 4), encoding (UTF-8), and file type (Java). The taskbar at the bottom includes icons for search, file explorer, and various applications like Google Chrome and Microsoft Word.

5. Write a program to print the below pattern

1

2 2

3 3 3

4 4 4 4

3 3 3

2 2

1

The screenshot shows a Java code editor in VS Code. The file NumberPattern.java contains the following code:

```
public class NumberPattern {
    public static void main(String[] args) {
        int n = 4;
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(i + " ");
            }
            System.out.println();
        }
        for (int i = n - 1; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print(i + " ");
            }
            System.out.println();
        }
    }
}
```

A yellow circular cursor is positioned at the start of the second outer loop's opening brace on line 10. The status bar at the bottom shows "Java: Warning".

6. Write a program to print hollow Square Dollar pattern?

A screenshot of a Java code editor interface. The top menu bar includes File, Edit, Selection, View, Go, and a search bar containing 'java'. The left sidebar shows an Explorer view with a JAVA folder containing 'HollowSquareDollarPattern.java'. The main editor area displays the following Java code:

```
1 public class HollowSquareDollarPattern {
2     public static void main(String[] args) {
3         int size = 5;
4         for (int i = 0; i < size; i++) {
5             for (int j = 0; j < size; j++) {
6                 if (i == 0 || i == size - 1 || j == 0 || j == size - 1) {
7                     System.out.print("$ ");
8                 } else {
9                     System.out.print(" ");
10                }
11            }
12        }
13    }
14 }
15 }
```

The code prints a hollow square pattern of '\$' characters with spaces in between. The status bar at the bottom shows 'Ln 3, Col 23' and other system information.

7. Write a program to print inverted pyramid pattern.

Input: no of rows: 3

Output

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

The screenshot shows a Java development environment with the following details:

- File Bar:** File, Edit, Selection, View, Go, ...
- Toolbar:** Run, Stop, Refresh, Minimize, Maximize, Close.
- Search Bar:** java
- Explorer:** Shows a file named "InvertedPyramidPattern.java" under the "JAVA" section.
- Code Editor:** Displays the following Java code:

```
1 public class InvertedPyramidPattern {  
2     public static void main(String[] args) {  
3         int rows = ;  
4         for (int i = 0; i < rows; i++) {  
5             for (int j = 0; j < i; j++) {  
6                 System.out.print(" ");  
7             }  
8             for (int k = 0; k < (2 * (rows - i) - 1); k++) {  
9                 System.out.print("*");  
10            }  
11        }  
12    }  
13}  
14}
```
- Bottom Status Bar:** Ln 3, Col 20, Spaces: 4, UTF-8, CRLF, Java, 08:24, 10-07-2024.

8. Write a program to reverse a number using loop?(Get the input from user)

Sample Input:

Number: 14567

Sample Output:

Reverse Number: 76541

Test cases:

1.-45721

2.000

3.AD1947

4.!@#\$%

5. $145 * 999 = 144855$

The screenshot shows a Java development environment. In the center is a code editor with the file `ReverseNumber.java` open. The code defines a class `ReverseNumber` with a `main` method that prints the reverse of the number 14567. Below the code editor is a terminal window displaying the command to run the Java executable and the resulting output: "Reverse Number: 76541". At the bottom of the screen is a Windows-style taskbar with various icons and system status indicators.

```
public class ReverseNumber {
    public static void main(String[] args) {
        int number = 14567;
        int reverse = 0;
        while (number != 0) {
            int digit = number % 10;
            reverse = reverse * 10 + digit;
            number /= 10;
        }
        System.out.println("Reverse Number: " + reverse);
    }
}
```

```
PS C:\Users\madet\OneDrive\Desktop\java> & 'C:\Program Files\Java\jre-1.8\bin\java.exe' '-cp' 'C:\Users\madet\AppData\Roaming\Code\User\workspaceStorage\3d6ae272a81ab392ab799090cdee6162\redhat.java\jdt_ws\java_80cf0e31\bin' 'ReverseNumber'
Reverse Number: 76541
PS C:\Users\madet\OneDrive\Desktop\java>
```

9. Write a program to convert the given decimal to binary and print the reverse of the binary decimal.

Input: 11

Output: 13

Explanation: $(11)_{10} = (1011)_2$.

After reversing the bits we get:

$(1101)_2 = (13)_{10}$.

Test cases:

1. 25
2. Eighteen
3. 12
4. -18

5. 34.5

The screenshot shows a Java code editor interface. The left sidebar has a tree view under 'EXPLORER' labeled 'JAVA' with a single file 'DecimalToBinaryReverse.java'. The main editor area contains the following code:

```
DecimalToBinaryReverse.java
DecimalToBinaryReverse.java > DecimalToBinaryReverse > main(String[])
1 public class DecimalToBinaryReverse {
2     public static void main(String[] args) {
3         int number = 11;
4         String binaryString = Integer.toBinaryString(number);
5         System.out.println("Binary representation: " + binaryString);
6         String reversedBinaryString = new StringBuilder(binaryString).reverse().toString();
7         System.out.println("Reversed binary representation: " + reversedBinaryString);
8         int reversedNumber = Integer.parseInt(reversedBinaryString, 2);
9         System.out.println("Decimal of reversed binary: " + reversedNumber);
10    }
11 }
```

Below the code, the terminal window shows the output of running the program:

```
PS C:\Users\madet\Desktop\java> & 'C:\Program Files\Java\jre-1.8\bin\java.exe' '-cp' 'C:\Users\madet\AppData\Roaming\Code\User\workspaceStorage\3d6ae272a81ab392ab799090cdee6162\redhat\java\jdt_ws\java_80cf0e31\bin' 'DecimalToBinaryReverse'
Binary representation: 1011
Reversed binary representation: 1101
Decimal of reversed binary: 13
PS C:\Users\madet\Desktop\java>
```

The status bar at the bottom right shows the date and time: 10-07-2024 08:27.

10. Write a program to find whether the person is eligible for vote or not. And if that particular person is not eligible, then print how many years are left to be eligible.

Sample Input:

Enter your age: 7

Sample output:

You are allowed to vote after 11 years

Test cases:

1. 25
2. Eighteen
3. 12
4. -18
5. 34.5

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named ".vscode" and a file named "VotingEligibility.java".
- Editor:** Displays the Java code for "VotingEligibility.java".

```
1 public class VotingEligibility {
2     public static void main(String[] args) {
3         int age = 7;
4         int votingAge = 18;
5         if (age >= votingAge) {
6             System.out.println("You are eligible to vote.");
7         } else {
8             int yearsLeft = votingAge - age;
9             System.out.println("You are allowed to vote after " + yearsLeft + " years.");
10        }
11    }
}
```
- Terminal:** Shows the command-line output of running the Java program.

```
PS C:\Users\madet\OneDrive\Desktop\java> & 'C:\Program Files\Java\jre-1.8\bin\java.exe' '-cp' 'C:\Users\madet\AppData\Roaming\Code\User\workspaceStorage\d6ae272a81ab392ab799900cdee6162\redhat\java\jdt_ws\java_80cf0e31\bin' 'VotingEligibility'
You are allowed to vote after 11 years.
PS C:\Users\madet\OneDrive\Desktop\java>
```
- Bottom Status Bar:** Shows the current file is "Java: Warning", line 4, column 29, spaces 4, encoding UTF-8, and CRLF.

ASSIGNMENT:4

1. Find the LCM and GCD of n numbers?

Sample Input:

N value = 2

Number 1 = 16

Number 2 = 20

Sample Output:

LCM = 80

GCD = 4

Test cases:

1. N = 3, {12, 25, 30}
2. N = 2, {52, 25, 63}
3. N = 3, {17, 19, 11}

4. $N = -2, \{52, 60\}$

5. $N = 2, \{30, 45\}$

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, ...
- Search Bar:** java
- Explorer:** JAVA folder containing .vscode and LCMGCD.java
- Editor:** LCMGCD.java file open, showing Java code for calculating LCM and GCD of an array.
- Bottom Status Bar:** Ln 27, Col 6, Spaces: 4, UTF-8, CRLF, Java
- Taskbar:** Shows various application icons.
- System Tray:** Shows date (10-07-2024), time (08:49), and system status (ENG IN).

```
LCMGCD.java 1  java_80cf0e31
LCMGCD.java > LCMGCD > gcdOfArray(int[])
1 public class LCMGCD {
2     static int lcmOfArray(int[] arr) {
3         int result = arr[0];
4         for (int i = 1; i < arr.length; i++) {
5             result = gcd(result, arr[i]);
6         }
7         return result;
8     }
9     static int gcdOfArray(int[] arr) {
10        int result = arr[0];
11        for (int i = 1; i < arr.length; i++) {
12            result = gcd(result, arr[i]);
13        }
14        return result;
15    }
16    public static void main(String[] args) {
17        int[] arr = {4, 6, 8, 12, 16};
18        int lcmResult = lcmOfArray(arr);
19        int gcdResult = gcdOfArray(arr);
20        System.out.println("LCM of the array elements: " + lcmResult);
21        System.out.println("GCD of the array elements: " + gcdResult);
22    }
23 }
```

2. Write a program using function to calculate the simple interest. Suppose the customer is a senior citizen. He is being offered 12 percent rate of interest; for all other customers, the ROI is 10 percent.

Sample Input:

Enter the principal amount: 200000

Enter the no of years: 3

Is customer senior citizen (y/n): n

Sample Output:

Interest: 60000

Test Cases:

1. Principal: 2000 , Years: 0

2. Principal: 20000 , Years: -2
3. Principal: -2000 , Years: 2
4. Principal: 2 , Years: 2000
5. Principal: 0 , Years: 5

The screenshot shows the Visual Studio Code interface. The left sidebar has 'EXPLORER' and 'JAVA' sections. The main area is a code editor with the file 'simpleintrest.java' open. The code calculates simple interest:

```

1 import java.util.Scanner;
2 public class simpleintrest{
3     public static void main (String[] args)
4     {
5         float p,r,t,si;
6         p=20000;r=12;t=3;
7         si=(p*r*t)/100;
8         System.out.println("simple intrest:"+si);
9     }
10 }
```

Below the code editor is a terminal window showing the command-line output of running the Java program:

```

PS C:\Users\madet\OneDrive\Desktop\java> java_80cf0e31
PS C:\Users\madet\OneDrive\Desktop\java> c:; cd 'c:\Users\madet\OneDrive\Desktop\java'; & 'C:\Program Files\Java\jre-1.8\bin\java.exe' '-cp' 'C:\Users\madet\AppData\Roaming\Code\User\workspaceStorage\3d6ae272a81ab392ab799090cdee6162\redhat.java\jdt_ws\java_80cf0e31\bin' 'simpleintrest'
simple intrest:72000.0
PS C:\Users\madet\OneDrive\Desktop\java> ^C
PS C:\Users\madet\OneDrive\Desktop\java> c:; cd 'c:\Users\madet\OneDrive\Desktop\java'; & 'C:\Program Files\Java\jre-1.8\bin\java.exe' '-cp' 'C:\Users\madet\AppData\Roaming\Code\User\workspaceStorage\3d6ae272a81ab392ab799090cdee6162\redhat.java\jdt_ws\java_80cf0e31\bin' 'simpleintrest'
simple intrest:72000.0
PS C:\Users\madet\OneDrive\Desktop\java>
```

The status bar at the bottom shows the file is 'Java: Warning'. The taskbar at the bottom includes icons for various applications like File Explorer, Task View, and Edge.

3. Write a program to print the Fibonacci series.

Sample Input:

Enter the n value: 6

Sample Output:

0 1 1 2 3 5

A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a file tree with a 'JAVA' folder containing '.vscode' and 'fibonacci.java'. The main editor tab is 'fibonacci.java 1' with the file path 'java_80cf0e31'. The code is a Java program to print the first 10 terms of the Fibonacci series:

```
fibonacci.java 1
fibonacci.java > fibonacci > main(String[])
1 import java.util.Scanner;
2 public class fibonacci{
3     public static void main(String[] args)      //here a1=first term; a2=second term; b=next te
4     {
5         int n=10,a1=0,a2=1;
6         System.out.println("fibonacci series:" + " " + "terms:");
7         for(int i=1;i<=n;i++){
8             System.out.println(a1+"");
9             int b=a1+a2;
10            a1=a2;
11            a2=b;
12        }
13    }
14 }
```

The status bar at the bottom shows 'Ln 6, Col 46' and other system information like battery level and network status.

4. Java Program to Find Even Sum of Fibonacci Series Till number N?

Sample Input: $n = 4$

Sample Output: 33

($N = 4$, So here the fibonacci series will be produced from 0th term till 8th term: 0, 1, 1, 2, 3, 5, 8, 13, 21)

Sum of numbers at even indexes = $0 + 1 + 3 + 8 + 21 = 33$)

The screenshot shows the Microsoft Visual Studio Code interface. The left sidebar has 'EXPLORER' expanded, showing 'JAVA' and a file named 'EvenSumFibonacci.java'. The main editor area displays the following Java code:

```
EvenSumFibonacci.java 1  java_80cf0e31
J EvenSumFibonacci.java > EvenSumFibonacci > main(String[])
1 import java.util.Scanner;
2 public class EvenSumFibonacci {
3     public static void main(String[] args) {
4         int n = 10;
5         if (n <= 0) {
6             System.out.println("Invalid input.");
7             return;
8         }
9         int[] fibonacci = new int[n + 1];
10        fibonacci[0] = 0;
11        fibonacci[1] = 1;
12        for (int i = 2; i <= n; i++) {
13            fibonacci[i] = fibonacci[i - 1] + fibonacci[i - 2];
14        }
15        int evenSum = 0;
16        for (int i = 0; i <= n; i += 2) {
17            evenSum += fibonacci[i];
18        }
19        System.out.println("Sum of numbers at even indexes in Fibonacci series up to term "
+ n + ": " + evenSum);
20    }
21
22
23 }
```

The status bar at the bottom indicates 'Ln 6, Col 46' and 'Spaces: 4'. The taskbar at the bottom shows various application icons.

5. Write a program to print the numbers from M to N by skipping K numbers in between?

Sample Input:

M = 50

N = 100

K = 7

Sample Output:

50, 58, 66, 74, ...

Test cases:

1. M = 15, N = 05, K = 02
2. M = 25, N = 50, K = 04
3. M = 15, N = 100, K = -02
4. M = 0, N = 0, K = 2

5. M = 200 , N = 200 , K = 5

6. Write a program to print all the composite numbers between a and b?

Sample Input:

A = 12

B = 19

Sample Output

14, 15, 16, 18

Test cases:

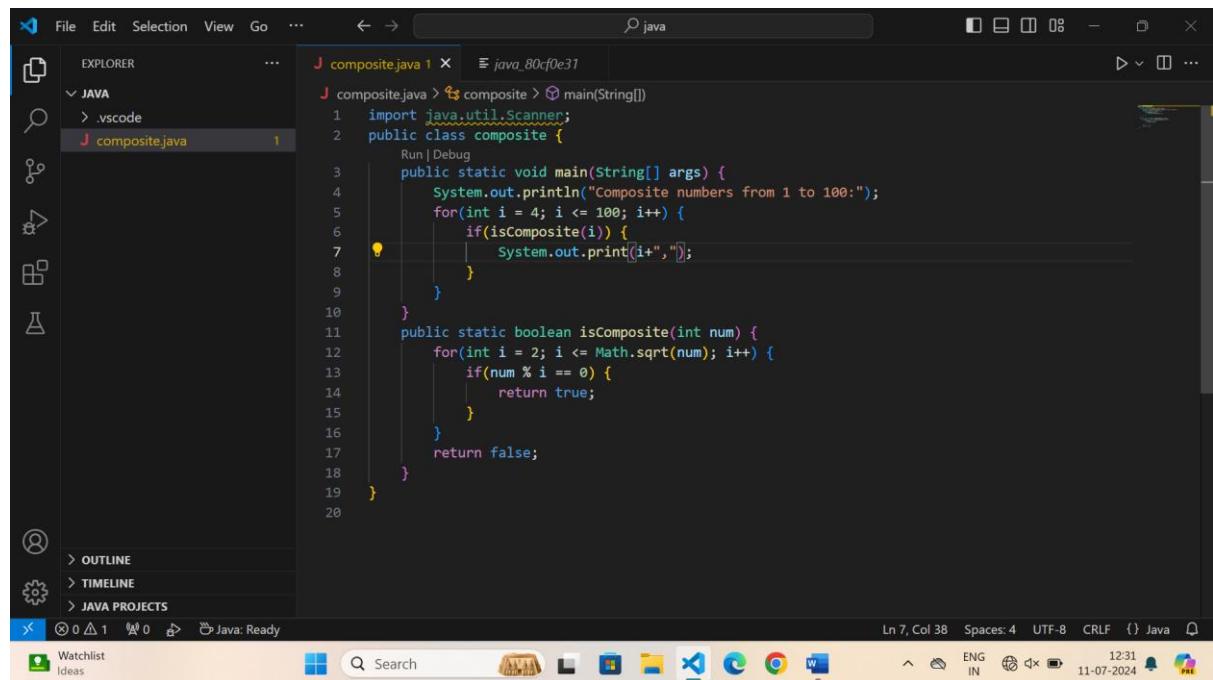
1. A = 11, B = 11

2. A = 20, B = 10

3. A = 0, B = 0

4. A = -5, B = 5

5. A = 7, B = -12



The screenshot shows the Visual Studio Code interface with a Java file named "composite.java" open in the editor. The code prints composite numbers from 1 to 100. It includes a helper method to check if a number is composite. The code is as follows:

```
import java.util.Scanner;
public class composite {
    public static void main(String[] args) {
        System.out.println("Composite numbers from 1 to 100:");
        for(int i = 4; i <= 100; i++) {
            if(isComposite(i)) {
                System.out.print(i+",");
            }
        }
    }
    public static boolean isComposite(int num) {
        for(int i = 2; i <= Math.sqrt(num); i++) {
            if(num % i == 0) {
                return true;
            }
        }
        return false;
    }
}
```

The code editor has syntax highlighting for Java, and the status bar at the bottom shows the file is Java: Ready.

7. Find the factorial of n?

Sample Input:

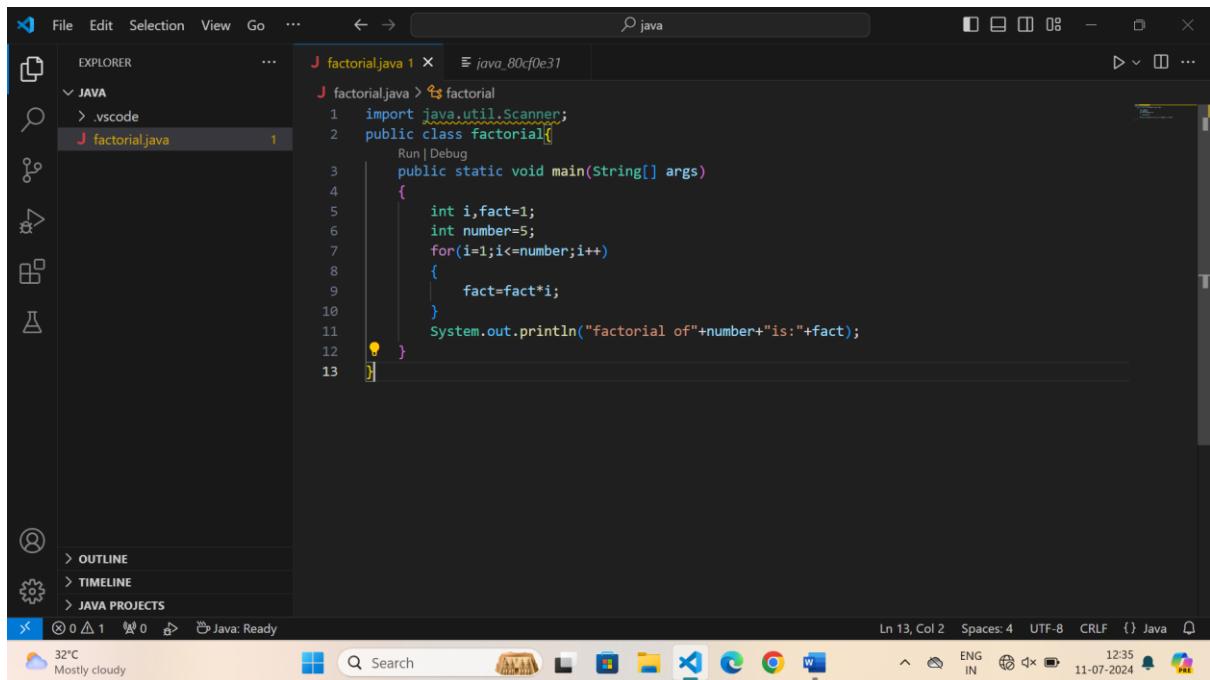
N = 4

Sample Output:

4 Factorial = 24

Test cases:

1. N = 0
2. N = -5
3. N = 1
4. N = Q
5. N = 3A



A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows the 'EXPLORER' view with a 'JAVA' folder containing '.vscode' and 'factorial.java'. The main editor tab is titled 'factorial.java 1' and contains the following Java code:

```
factorial.java > factorial
1 import java.util.Scanner;
2 public class factorial{
3     public static void main(String[] args)
4     {
5         int i,fact=1;
6         int number=5;
7         for(i=1;i<=number;i++)
8         {
9             fact=fact*i;
10        }
11        System.out.println("factorial of"+number+"is:"+fact);
12    }
13 }
```

The status bar at the bottom shows 'Ln 13, Col 2' and other system information like weather and date.

8. Find the year of the given date is leap year or not

Sample Input:

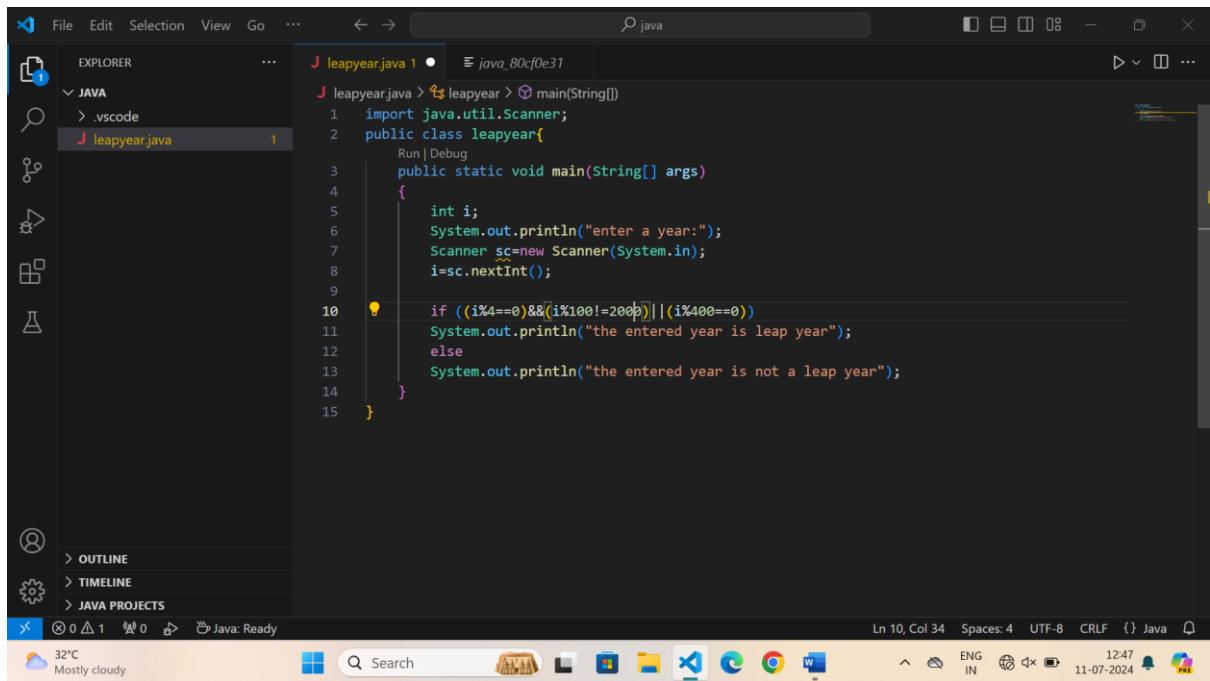
Enter Date: 04/11/1947

Sample Output:

Given year is Non Leap Year

Test cases:

1. 04/11/1947
2. 11/15/1936
3. 31/45/1996
4. 64/09/1947
5. 00/00/2000



```
leapyear.java 1 ● java_80cf0e31
J leapyear.java > leapyear > main(String[])
1 import java.util.Scanner;
2 public class leapyear{
3     public static void main(String[] args)
4     {
5         int i;
6         System.out.println("enter a year:");
7         Scanner sc=new Scanner(System.in);
8         i=sc.nextInt();
9
10        if ((i%4==0)&&(i%100!=200)|||(i%400==0))
11            System.out.println("the entered year is leap year");
12        else
13            System.out.println("the entered year is not a leap year");
14    }
15 }
```

9. Find the number of factors for the given number

Sample Input:

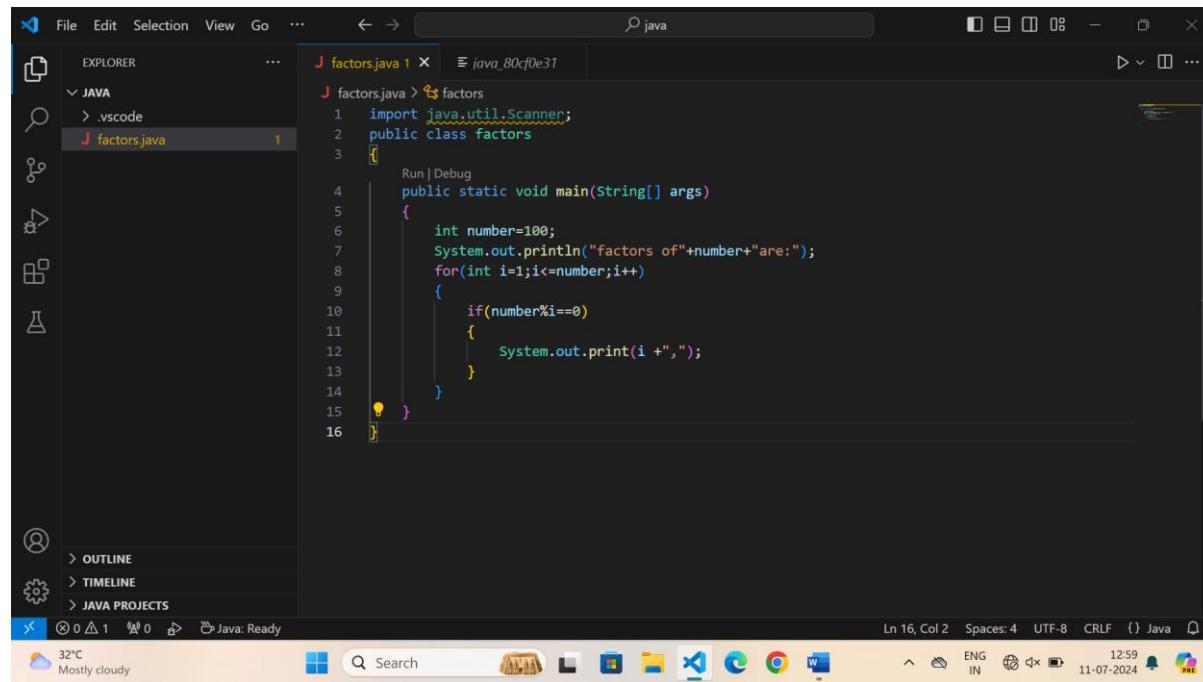
Given number: 100

Sample Output:

Number of factors = 9

Test cases:

1. 343
2. 1080
3. -243
4. 101010
5. 0



The screenshot shows the Visual Studio Code interface with a Java project open. The Explorer sidebar shows a folder named 'JAVA' containing '.vscode' and 'factors.java'. The 'factors.java' file is the active tab, displaying the following code:

```
import java.util.Scanner;
public class factors
{
    public static void main(String[] args)
    {
        int number=100;
        System.out.println("factors of "+number+" are:");
        for(int i=1;i<=number;i++)
        {
            if(number%i==0)
            {
                System.out.print(i+",");
            }
        }
    }
}
```

The code prints the factors of 100. The terminal tab at the bottom shows 'Java: Ready'. The status bar indicates 'Ln 16, Col 2'.

10. Write a program to print the given number is Perfect number or not?

Sample Input:

Given Number: 6

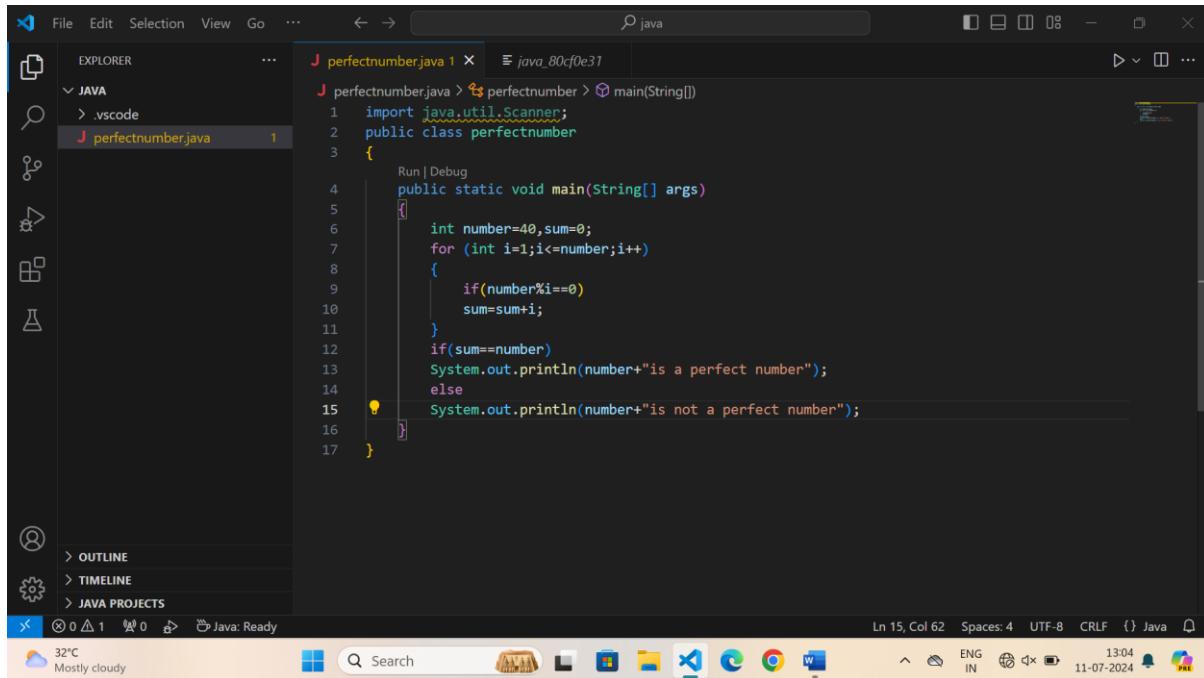
Sample Output:

It's a Perfect Number

Test cases:

1. 17
2. 26!
3. 143
4. 84.1

5. -963



A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows the Explorer panel with a 'JAVA' folder containing 'perfectnumber.java'. The main editor tab is titled 'perfectnumber.java 1' and contains the following Java code:

```
1 import java.util.Scanner;
2 public class perfectnumber
3 {
4     public static void main(String[] args)
5     {
6         int number=40,sum=0;
7         for (int i=1;i<=number;i++)
8         {
9             if(number%i==0)
10                 sum=sum+i;
11         }
12         if(sum==number)
13             System.out.println(number+"is a perfect number");
14         else
15             System.out.println(number+"is not a perfect number");
16     }
17 }
```

The status bar at the bottom indicates 'Ln 15, Col 62' and 'Java: Ready'. The system tray shows the date and time as '11-07-2024 13:04'.

ASSIGNMENT:5

1. Write a program to find the square, cube of the given decimal number

Sample Input:

Given Number: 0.6

Sample Output:

Square Number: 0.36

Cube Number: 0.216

```

Java - javaprogram/src/javaprogram/Square_cube.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer X
javaprogram
  src
    javaprogram
      age.java
      Alphabeticalorder.java
      compositebetween.java
      Counttowel.java
      decimal_to_binary.java
      fact.java
      factors.java
      fibinocci.java
      frequency.java
      Lcm_Gcd.java
      leapyear.java
      matrixaddition.java
      matrixmultiplication.java
      Mean_median_mode.java
      meammedianmode.java
      mergearray.java
      no_of_composite.java
      odd_numbers.java
      pascal_triangle.java
      prefect_number.java
      Printnewejava
      Removeeven.java
      Reverse_A_number.java
      Reversealphabet.java
      Right_triangle_star_pattern.java
      Simple_intrst.java
      skipping.java
      Specialcharacters.java
      Square_cube.java
      stringreverse.java
      Stringsearch.java
      stringToInteger.java
      testJava
      Validname.java
  stringTolite...
  Validname.java
  Alphabetical...
  Printvowel.java
  Square_cube...
  frequency.java
  test.java
Task List X
Outline X
javaprogram
  Square_cube
    main(String[])
Console X
Problems Javadoc Declaration Debug
<terminated> Square_cube [Java Application] C:\Users\ganes\p2\pool\plugins\org.eclipse.jdt.core\full\win32\x86_64_22.0.1.v20240426-1149\jre\bin\javaw.exe (13 Jul 2024, 12:29:11 pm) [1]
4 8

```

2. Find the n^{th} odd number after n odd number

Sample Input: N : 7

Sample Output:

Hence the values printed for i are 1 , 3 , 5.

```

Java - javaprogram/src/javaprogram/odd_numbers.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer X
javaprogram
  src
    javaprogram
      age.java
      Alphabeticalorder.java
      compositebetween.java
      Counttowel.java
      decimal_to_binary.java
      fact.java
      factors.java
      fibinocci.java
      frequency.java
      Lcm_Gcd.java
      leapyear.java
      matrixaddition.java
      matrixmultiplication.java
      Mean_median_mode.java
      meammedianmode.java
      mergearray.java
      no_of_composite.java
      odd_numbers.java
      pascal_triangle.java
      prefect_number.java
      Printnewejava
      Removeeven.java
      Reverse_A_number.java
      Reversealphabet.java
      Right_triangle_star_pattern.java
      Simple_intrst.java
      skipping.java
      Specialcharacters.java
      Square_cube.java
      stringreverse.java
      Stringsearch.java
      stringToInteger.java
      testJava
      Validname.java
  stringTolite...
  Validname.java
  Alphabetical...
  Printvowel.java
  Square_cube...
  frequency.java
  test.java
Task List X
Outline X
javaprogram
  odd_numbers
    main(String[])
Console X
Problems Javadoc Declaration Debug
<terminated> odd_numbers [Java Application] C:\Users\ganes\p2\pool\plugins\org.eclipse.jdt.core\full\win32\x86_64_22.0.1.v20240426-1149\jre\bin\javaw.exe (13 Jul 2024, 12:30:30 pm) [1]
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35

```

3. Program to find whether the given number is Armstrong number or not

Sample Input:

Enter number: 153

Sample Output:

Given number is Armstrong number

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with a package named "javaprogram" containing various Java files like age.java, Armstrong.java, etc.
- Code Editor:** Displays the content of the Armstrong.java file. The code reads a number from the user, checks if it's an Armstrong number, and prints the result.
- Console:** Shows the output of the program when run with the input "153". It prints "153 is an Armstrong number."
- Outline:** Shows the class structure with methods main() and isArmstrong().

4. Write a program to find the sum of digits of N digit number (sum should be single digit)

Sample Input:

Enter N value: 3

Enter 3 digit numbers: 143

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with a package named "javaprogram" containing various Java files like age.java, Armstrong.java, etc.
- Code Editor:** Displays the content of the sumofdigits.java file. The code reads a number from the user and calculates the sum of its digits.
- Console:** Shows the output of the program when run with the input "143". It prints "Sum of digits : 6".
- Outline:** Shows the class structure with methods main() and sumofdigits().

5. Write a program to find the square root of a perfect square number(print both the positive and negative values)

Sample Input:

Enter the number: 6561

Sample Output:

Square Root: 81, -81

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a Java project named "javaprogram" containing several source files like "squarerootofperfectnumber.java", "frequency.java", etc.
- Code Editor:** Displays the code for "squarerootofperfectnumber.java". The code reads a number from the user, calculates its square root, and prints it if it's a perfect square.
- Console:** Shows the output of running the program with the input "4", which prints "Square Root: 2, -2".
- Task List:** Shows a single task entry.
- Outline:** Shows the class structure of "javaprogram".
- Bottom Bar:** Shows system icons for battery, signal, and date/time (13-07-2024).

```

1 package javaprogram;
2 import java.util.Scanner;
3 public class squarerootofperfectnumber {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         int number = scanner.nextInt();
7         double sqrt = Math.sqrt(number);
8         if (sqrt - Math.floor(sqrt) == 0) {
9             System.out.println("Square Root: " + (int)sqrt + ", " + (-int)sqrt);
10        } else {
11            System.out.println("The entered number is not a perfect square.");
12        }
13    }
14 }
15 
```

6. Write a program to given an integer n, return true if it is a power of three. Otherwise, return false.

Input =27

Output= true

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a Java project named "javaprogram" containing several source files like "powerofthree.java", "frequency.java", etc.
- Code Editor:** Displays the code for "powerofthree.java". It uses a while loop to repeatedly divide the number by 3 until it's less than 1 or not divisible by 3, then checks if the result is 1.
- Console:** Shows the output of running the program with the input "27", which prints "false".
- Task List:** Shows a single task entry.
- Outline:** Shows the class structure of "javaprogram".
- Bottom Bar:** Shows system icons for battery, signal, and date/time (13-07-2024).

```

1 package javaprogram;
2 import java.util.Scanner;
3 public class powerofthree {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Enter an integer: ");
7         int n = scanner.nextInt();
8         boolean result = isPowerOfThree(n);
9         System.out.println(result);
10    }
11    public static boolean isPowerOfThree(int n) {
12        if (n < 1) {
13            return false;
14        }
15        while (n % 3 == 0) {
16            n /= 3;
17        }
18        return n == 1;
19    }
20 }
21 
```

7. Write a program to given a string paragraph and a string array of the banned words banned, return the most frequent word that is not banned. It is guaranteed there is at least one word that is not banned, and that the answer is unique.

Input Paragraph="Ram hit a ball, the hit ball flew far after it was hit",
Banned = [hit]

Output="Ball"

8. Write a program to given a fixed-length integer array arr, duplicate each occurrence of zero, shifting the remaining elements to the right.

Input: arr = [1, 0, 2, 3, 0, 4, 5, 0]

Output: [1, 0, 0, 2, 3, 0, 0, 4]

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Java - javaprogram/srccjavaprogram/duplicatezeros.java - Eclipse IDE
- Menu Bar:** File, Edit, Source, Refactor, Source, Navigate, Search, Project, Run, Window, Help
- Toolbars:** Standard toolbar with icons for New, Open, Save, Cut, Copy, Paste, Find, etc.
- Left Sidebar:** Package Explorer showing various Java files in the workspace.
- Central Area:** Code editor displaying the `DuplicateZeros` class with its main method and a static helper method `duplicateZeros(int[] arr)`.
- Bottom Status Bar:** Shows the current file path (`javaprogram/duplicatezeros.java`), line number (13), and column number (21).
- Bottom Icons:** Includes icons for Problems, Javadoc, Declaration, Console, and Debug.

9. Write a program to given an array `nums` containing n distinct numbers in the range $[0, n]$, return the only number in the range that is missing from the array.

Input nums = [3, 0, 1]

Output: 2

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Java - javaprogram/src/javaprogram/missingnumber.java - Eclipse IDE
- Package Explorer:** Shows various Java files including alphabeticalorder.java, Armstrong.java, and missingnumber.java.
- Code Editor:** Displays the `missingnumber.java` code:

```

1 package javaprogram;
2 import java.util.*;
3
4 public class missingnumber {
5     public static void main(String[] args) {
6         int[] nums = {1, 0, 1}; // Example Input
7         System.out.println("The missing number is: " + findMissingNumber(nums));
8     }
9     public static int findMissingNumber(int[] nums) {
10        int n = nums.length;
11        int expectedSum = n * (n + 1) / 2;
12        int actualSum = 0;
13
14        for (int num : nums) {
15            actualSum += num;
16        }
17        return expectedSum - actualSum;
18    }
19 }

```
- Console:** Shows the output: `The missing number is: 2`.
- Outline:** Shows the class structure: `javaprogram > missingnumber`.
- Bottom Status Bar:** Shows the date and time: 13-07-2024, 14:21.

10. Write a program to given an integer array nums, find the subarray with the largest sum, and return its sum.

Input `nums = [-2, 1, -3, 4, -1, 2, 1, -5, 4]`

Output: 6

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Java - javaprogram/src/javaprogram/largestsubarray.java - Eclipse IDE
- Package Explorer:** Shows various Java files including alphabeticalorder.java, Armstrong.java, and largestsubarray.java.
- Code Editor:** Displays the `largestsubarray.java` code:

```

1 package javaprogram;
2 public class largestsubarray {
3     public static void main(String[] args) {
4         int[] nums = {-2, 1, -3, 4, -1, 2, 1, -5, 4}; // Example input
5         System.out.println("The largest sum of a subarray is: " + maxSubArray(nums));
6     }
7     public static int maxSubArray(int[] nums) {
8         int currentSum = nums[0];
9         int maxSum = nums[0];
10
11        for (int i = 1; i < nums.length; i++) {
12            currentSum = Math.max(nums[i], currentSum + nums[i]);
13            maxSum = Math.max(maxSum, currentSum);
14        }
15
16        return maxSum;
17    }
18 }

```
- Console:** Shows the output: `The largest sum of a subarray is: 6`.
- Outline:** Shows the class structure: `javaprogram > largestsubarray`.
- Bottom Status Bar:** Shows the date and time: 13-07-2024, 14:22.