

SAVEETHA SCHOOL OF ENGINEERING
SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
INSTITUTE OF PLACEMENT AND TRAINING
CSA09 –JAVA PROGRAMMING

Name:V.Balaram Krishna
Reg.No:192324049

ASSIGNMENT - 1

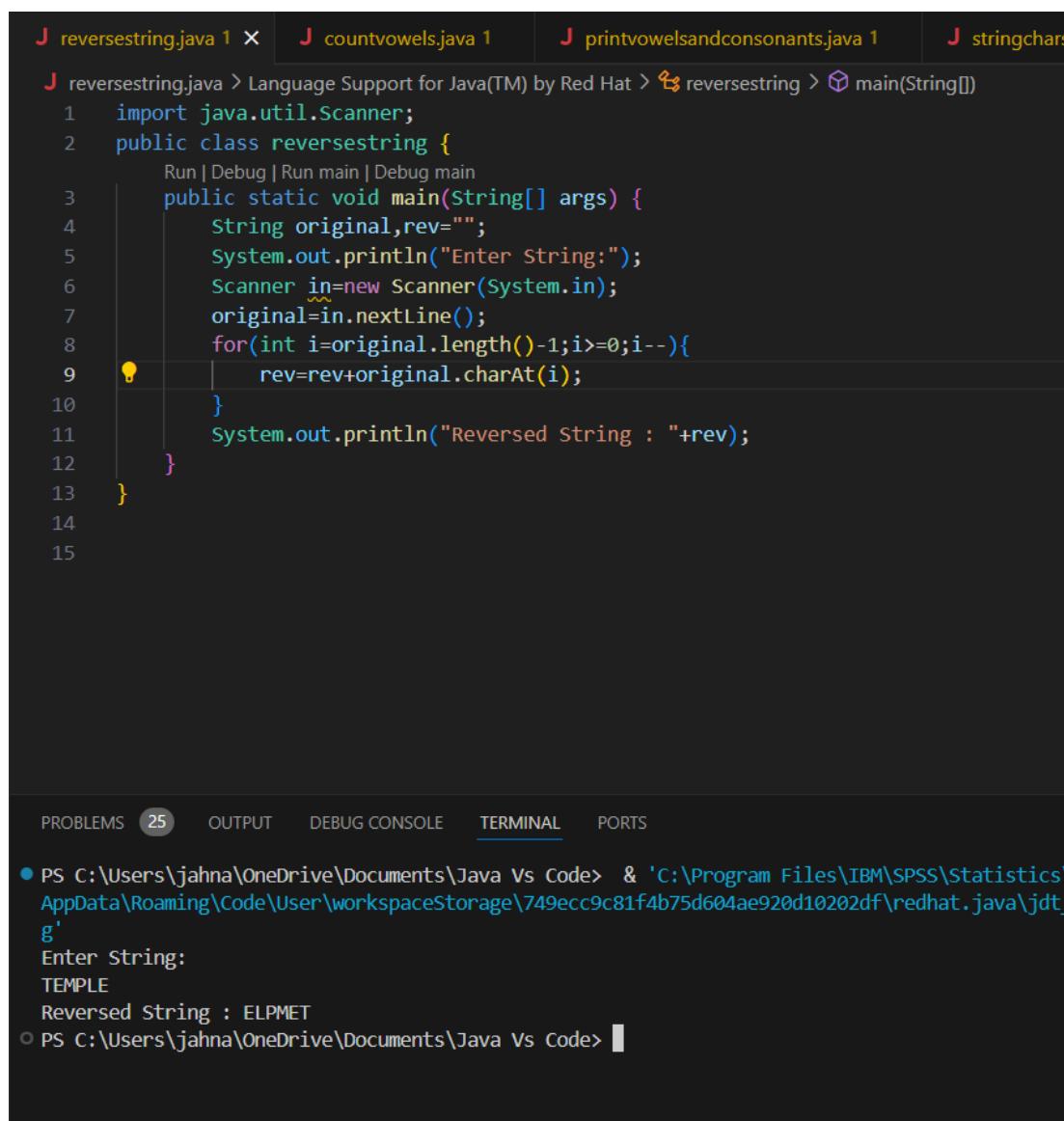
1. Write a program to reverse a word using loop? (Not to use inbuilt functions)

Sample Input:

String: TEMPLE

Sample Output:

Reverse String: ELPMET



```
J reversestring.java 1 X J countvowels.java 1 J printvowelsandconsonants.java 1 J stringchars
J reversestring.java > Language Support for Java(TM) by Red Hat > ⇧ reversestring > ⚡ main(String[])
1 import java.util.Scanner;
2 public class reversestring {
3     Run | Debug | Run main | Debug main
4     public static void main(String[] args) {
5         String original,rev="";
6         System.out.println("Enter String:");
7         Scanner in=new Scanner(System.in);
8         original=in.nextLine();
9         for(int i=original.length()-1;i>=0;i--){
10             rev=rev+original.charAt(i);
11         }
12         System.out.println("Reversed String : "+rev);
13     }
14 }
15
```

PROBLEMS 25 OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code & 'c:\Program Files\IBM\SPSS\Statistics\19\bin\java' -jar redhat.jar
- Enter String:
TEMPEL
Reversed String : ELPMET
- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code

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

Sample Input:

String: 1234

Sample Output:

Out put String: 1234

```
J stringtointeger.java > ...
1  public class stringtointeger {
2      Run | Debug | Run main | Debug main
3      public static void main(String[] args) {
4          String n="123";
5          System.out.println(Integer.parseInt(n));
6      }
7  }
```

PROBLEMS 25 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\...
AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\...
g'
Enter String:
TEMPLE
Reversed String : ELPMET
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> []
```

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

CODE:

The screenshot shows the Java Vs Code IDE interface. The code editor displays a Java file named `validuser.java` with the following content:

```
1  J validuser.java 1 x  J squareando... || ⏪ ⏴ ⏵ ⏶ ⏷ stringarangeletteralph... interest.java 2  J generalfib.java 1  J leapyear.java 1  J arraypry... D ...  
J validuser.java > Language Support for Java(TM) by Red Hat > validuser > main(String[])  
1 import java.util.Scanner;  
2 public class validuser {  
3     Run | Debug | Run main | Debug main  
4         String user;  
5         Scanner a=new Scanner(System.in);  
6         System.out.println("Enter Username : ");  
7         user=a.nextLine();  
8         String res="^[_A-Z-a-z-0-9_]{0,17}$";  
9         if(user.matches(res))  
10             System.out.println("Valid Username");  
11         else  
12             System.out.println("Not valid");  
13     }  
14 }
```

The terminal pane at the bottom shows the following output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ^ x  
2 ● PS C:\Users\jahnna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'c:\Users\jahnna\AppData\Roaming\Code\User\workspaceStorage\749ec9c81fb75d694ae92d10282df\redhat\java\jdt_ws\Java Vs Code_8e18c08a\bin' 'validuser'  
1 Enter Username :  
1 balu 456  
1 Valid Username  
0 PS C:\Users\jahnna\OneDrive\Documents\Java Vs Code>
```

A dropdown menu on the right lists various run configurations:

- Run: general
- Run: generate
- Run: genera...
- Run: gener...
- Run: compo...
- Run: factorial
- Run: leapyear
- Run: factors
- Run: perfect...
- Run: validus...
- Run: validus...

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

FF

Order(A/D) : A

Sample Output:

Apple

--PP--
Banana

Carrot

Jack

Radish

OLITERI

OUTPUT

J sortlistalphabet.java > Language Support for Java(TM) by Red Hat > sortlistalphabet > main(String[])

```
1  public class sortlistalphabet {  
2      Run | Debug | Run main | Debug main  
3      public static void main(String[] args) {  
4          int n=3;  
5          String arr[]={ "Ajay", "Balu", "Aravind" };  
6          String temp;  
7  
8          for(int i=0;i<n;i++){  
9              for(int j=i+1;j<n;j++){  
10                  if (arr[i].compareTo(arr[j])>0){  
11                      temp=arr[i];  
12                      arr[i]=arr[j];  
13                      arr[j]=temp;  
14                  }  
15              }  
16              System.out.println(arr[i]);  
17          }  
18      }  
19  }  
20  
21
```

PROBLEMS 25 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java -Djava.ext.dirs=%JAVA_HOME%\lib\ext -jar %~dp0\redhat.java\jdt_ws\Java Vabets'\sortlistalphabet
Ajay
Aravind
Balu
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

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

OUTPUT:

```

J special.java > Language Support for Java(TM) by Red Hat > 🛡 special > ⚒ main(String[])
1  public class special {
2      Run | Debug | Run main | Debug main
3      public static void main(String[] args) {
4          String sentence = "hello @456%";
5
6          for (int i = 0; i < sentence.length(); i++) {
7              char ch = sentence.charAt(i);
8              if (!Character.isLetterOrDigit(ch) && !Character.isWhitespace(ch)) {
9                  System.out.print(ch);
10             }
11         }
12     }
13

```

PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'special' @%
- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

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

OUTPUT:

```

J countvowels.java > Language Support for Java(TM) by Red Hat > 🛡 countvowels > ⚒ main(String[])
1  import java.util.Scanner;
2  public class countvowels {
3      Run | Debug | Run main | Debug main
4      public static void main(String[] args) {
5          String original;
6          int count=0;
7          System.out.print("Enter String ");
8          Scanner in=new Scanner(system.in);
9          original=in.nextLine();
10         original=original.toLowerCase();
11         int len=original.length();
12         for(int i=0;i<len;i++){
13             if(original.charAt(i)=='a'||original.charAt(i)=='e'||original.charAt(i)=='i'||original.charAt(i)=='o'||original.charAt(i)
14                 =='u'){
15                 count+=1;
16             }
17         }
18     }

```

PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'countvowels'

Enter String hello
No of Vowels :2

PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

- 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

OUTPUT:

```
J printvowelsandconsonants.java > ...
1 import java.util.Scanner;
2 class printvowelsandconsonants{
3     Run | Debug | Run main | Debug main
4     public static void main(String[] args) {
5         String original;
6         String vowels="aeiou";
7         String Con="";
8         System.out.println("Enter String: ");
9         Scanner in=new Scanner(System.in);
10        original=in.nextLine();
11        original=original.toLowerCase();
12        int n=original.length();
13        System.out.print("Vowels :");
14        for(int i=0;i<n;i++){
15            char ch=original.charAt(i);
16            if(vowels.indexOf(ch)!=-1){
17                System.out.print(ch+" ");
18            } else if(Character.isLetter(ch)){
19                Con+=ch+" ";
20            }
21        }
22        System.out.println("\nConsonants "+Con);
23    }
24 }
```

PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> ^C
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> c:; cd 'c:\Users\jahna\OneDrive\Documents\Java Vs Code'; & 'C:\Programs\Statistics25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604aedhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'printvowelsandconsonants'
Enter String:
saveetha
Vowels :a e e a
Consonants s v t h
```

- 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.

OUTPUT:

```

1 import java.util.Scanner;
2 public class stringsearch {
3     public static void main(String[] args) {
4         String str;
5         int found=-1;
6         String p;
7         System.out.println("Enter String :");
8         Scanner n=new Scanner(System.in);
9         str=n.nextLine();
10        System.out.println("Enter Element You want to search:");
11        p=n.nextLine();
12        str=str.toLowerCase();
13        p=p.toLowerCase();
14        for(int i=0;i<str.length();i++){
15            if(p.equals(str.valueOf(str.charAt(i)))) {
16                found=i;
17                break;
18            }
19        }
20        System.out.println("Element found at index : "+found);
21    }
22 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\jaha\OneDrive\Documents\Java Vs Code & 'C:\Program Files\IBM\SPSS\statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jaha\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat\java\jdt_ws\Java Vs Code_8e18c80a\bin' 'stringcharsearch'
Enter String :
HELLO WORLD
Enter Element You want to search:
W
Element found at index : 6
PS C:\Users\jaha\OneDrive\Documents\Java Vs Code

av: Ready

ln 20, Col 24 Spaces: 4 UTF-8 CRLF Java

powershell Run: reverse... Run: stringa... Run: sortlist... Run: count... Run: printvo... Run: stringc...

12:02 13-07-2024

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

OUTPUT:

The screenshot shows the Java Vs Code interface with several tabs open:

- `countvowels.java`
- `printvowels.java`
- `stringarrangelettersalphareverse.java`
- `stringarrangelettersalphareverse.java` (active tab)
- `stringremovevowels.java`

The active tab contains the following Java code for reversing a string and arranging letters alphabetically:

```

1 public class stringarrangelettersalphareverse {
2     public static void main(String[] args) {
3         System.out.println("Enter String :");
4         Scanner sc=new Scanner(System.in);
5         str=sc.nextLine();
6         char[] arr=str.toCharArray();
7         for(int i=0;i<arr.length;i++){
8             for (int j=i;j<arr.length;j++){
9                 if (arr[i]>arr[j]) {
10                     char temp=arr[i];
11                     arr[i]=arr[j];
12                     arr[j]=temp;
13                 }
14             }
15         }
16         System.out.print("Alphabetical Order : ");
17         for(int i=str.length()-1;i>=0;i--){
18             System.out.print(arr[i]);
19         }
20     }
21 }
22
23
24
25

```

The terminal below shows the execution of the code and its output:

```

PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> ^C
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> cd 'c:\Users\jahna\OneDrive\Documents\Java Vs Code' & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81fb75d604ae920d10202df\redhat.java\dt_ws\Java Vs Code_8e18c80a\bin' 'stringarrangelettersalphareverse'
Enter String :
MOSQUE
Alphabetical Order : USQOME
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

```

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 y thgm

The screenshot shows the Java Vs Code interface with several tabs open:

- `countvowels.java`
- `printvowels.java`
- `stringarrangelettersalphareverse.java`
- `stringremovevowels.java` (active tab)

The active tab contains the following Java code for removing vowels from a sentence:

```

1 import java.util.Scanner;
2 public class stringremovevowels {
3     public static void main(String[] args) {
4         String str;
5         Scanner n=new Scanner(System.in);
6         System.out.print("Enter Sentence : ");
7         str=n.nextLine();
8         String nonvowels=str.replaceAll("[aeiouAEIOU]","");
9         System.out.println("Result");
10        System.out.print(nonvowels);
11    }
12
13
14

```

The terminal below shows the execution of the code and its output:

```

PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81fb75d604ae920d10202df\redhat.java\dt_ws\Java Vs Code_8e18c80a\bin' 'stringremovevowels'
Enter Sentence : WE CAN PLAY GAME
Result
W C N PL Y G M
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

```

ASSIGNMENT-2

11. 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

OUTPUT:

The screenshot shows the Java Vs Code interface. The left pane displays the code for 'arraysmatrixmul.java'. The right pane shows the terminal output of the program's execution.

```
1 public class arraysmatrixmul {
2     public static void main(String[] args){
3         int[][] mat1={{1,2,3},{3,4,5},{1,1,1}};
4         int[][] mat2={{1,1,1},{1,1,1},{1,1,1}};
5         int[][] result=new int[mat1.length][mat2[0].length];
6         for(int i=0;i<mat1.length;i++){
7             for(int j=0;j<mat1[0].length;j++){
8                 for(int k=0;k<mat2[0].length;k++){
9                     result[i][j]+=mat1[i][k]*mat2[k][j];
10                }
11            }
12        }
13        for(int i=0;i<mat1.length;i++){
14            for(int j=0;j<mat2[0].length;j++){
15                System.out.print(result[i][j]+" ");
16            }
17            System.out.println();
18        }
19    }
20 }
```

Terminal Output:

```
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'c:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc0c81f4b75d604ae920d10202f\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'arraysmatrixmul'
6 6 6
12 12 12
3 3 3
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
```

12. 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

OUTPUT:

```

1  public class arraysmatrixadd {
2      Run | Debug | Run main | Debug main
3      public static void main(String[] args){
4          int[][] mat1={{1,1,1},{1,1,1},{1,1,1}};
5          int[][] mat2={{1,1,1},{1,1,1},{1,1,1}};
6          int[][] result=new int[mat1.length][mat2[0].length];
7          for(int i=0;i<mat1.length;i++){
8              for(int j=0;j<mat2[0].length;j++){
9                  result[i][j]=mat1[i][j]+mat2[i][j];
10             }
11         }
12         for(int i=0;i<mat1.length;i++){
13             for(int j=0;j<mat2[0].length;j++){
14                 System.out.print(result[i][j]+" ");
15             }
16         }
17     }
18 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'arraysmatrixadd'
 2 2 2
2 2 2
2 2 2
- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

java: Ready

Ln 10, Col 10 Spaces: 4 UTF-8 CRLF 12:05 13-07-2024

13. 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}

OUTPUT:

```

1  public class arraymergesorted {
2      Run | Debug | Run main | Debug main
3      public static void main(String[] args){
4          int[] arr1={1,2,3,4};
5          int[] arr2={2,4,5,6};
6          int n1=arr1.length;
7          int n2=arr2.length;
8          int[] arr3=new int[n1+n2];
9          System.out.println("Mergesorted Array:");
10         mergearrays(arr1,arr2,n1,n2,arr3);
11         for(int i=0;i<n1+n2;i++){
12             System.out.print(arr3[i]+" ");
13         }
14     }
15     public static void mergearrays(int[] arr1,int[] arr2,int n1,int n2,int[] arr3){
16         int i=0,j=0,k=0;
17         while(i<n1&&j<n2){
18             if(arr1[i]<arr2[j]){
19                 arr3[k++]=arr1[i++];
20             }
21             else{
22                 arr3[k++]=arr2[j++];
23             }
24         }
25     }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'arraymergesorted'
 Mergesorted Array:
1 2 2 3 4 4 5 6
- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

java: Ready

Ln 9, Col 41 Spaces: 4 UTF-8 CRLF 12:06 13-07-2024

14. 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}

CODE:

```
J arraysmeanmedian.java > Language Support for Java(TM) by Red Hat > arraysmeanmedian
1   import java.util.Arrays;
2
3   public class arraysmeanmedian {
4       Run | Debug | Run main | Debug main
5       public static void main(String[] args) {
6           int[] arr={16, 18, 27, 16, 23, 21, 19};
7           int n=arr.length;
8           System.out.print("Mean :");
9           mean(arr,n);
10          System.out.print("Median :");
11          median(arr,n);
12          System.out.print("Mode :");
13          mode(arr,n);
14      }
15      public static void mean(int[] arr,int n){
16          int Sum=0;
17
18          for(int i=0;i<n;i++){
19              Sum=Sum+arr[i];
20          }
21          System.out.println(Sum/n);
22      }
23      public static void median(int[] arr,int n){
24          Arrays.sort(arr);
25          System.out.println(arr[n/2]);
26      }
27      public static void mode(int[] arr,int n){
28          Arrays.sort(arr);
29
30          for(int i=0;i<n;i++){
31              if(arr[i]==arr[i+1]){
32                  System.out.println(arr[i]);
33                  break;
34              }
35          }
36      }
}
```

OUTPUT:

```
Mean :20
Median :19
Mode :16
```

15. 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 }

OUTPUT:

The screenshot shows the Java Vs Code IDE interface. The code editor tab bar has several files: perfectsquare.java, arraycomposite.java (which is the active file), stringarrangelettersalph, invalid.java, arraysmeanmedian.java, and arraycomposite.java again. The arraycomposite.java file contains Java code to count composite numbers in an array. The terminal tab shows the command line output of running the program with the input array [16, 18, 27, 16, 23, 21, 19], resulting in the output "5". Below the terminal is the Windows taskbar with various application icons.

```
public class arraycomposite {
    public static void main(String[] args) {
        int[] arr={16,18,27,16,23,21,19};
        int Count=0;
        for(int i=0;i<arr.length;i++){
            if(compo(arr[i])){
                Count+=1;
            }
        }
        System.out.println(Count);
    }
    public static boolean compo(int num){
        if(num<2){
            return false;
        }
        for(int i=2;i<=Math.sqrt(num);i++){
            if(num%i==0){
                return true;
            }
        }
        return false;
    }
}
```

PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:/Users/jahna/AppData\Roaming\Code/User/workspaceStorage/749ecc9c81fab75d604ae920d10202df/redhat-.java/jdt_ws/Java Vs Code_8e18c80a/bin' 'arraycomposite'
5
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

Patterns :

16. Write a program to print Right Triangle Star Pattern

Sample Input:: n = 5

Output:

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

OUTPUT:

The screenshot shows the Java Vs Code IDE interface. The code editor at the top contains a Java program named `arraypyramid.java`. The terminal below it displays the output of running the program, which prints a pyramid pattern of asterisks. The status bar at the bottom shows the current file path, line number, and date.

```
arraypyramid.java 1 > Language Support for Java(TM) by Red Hat > arraypyramid
1 import java.util.Scanner;
2 public class arraypyramid {
3     public static void main(String[] args){
4         int n;
5         Scanner a=new Scanner(System.in);
6         System.out.println("Enter Rows:");
7         n=a.nextInt();
8         for(int i=0;i<n;i++){
9             for(int j=0;j<n-i;j++){
10                 System.out.print(" ");
11             }
12             for(int k=0;k<=i;k++){
13                 System.out.print("* ");
14             }
15             System.out.println();
16         }
17     }
18 }
```

PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ouming\Code\User\workspacestorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\java_vs_code_8e18c8ea\bin\arraypyramid
Enter Rows:
5
*
* *
* * *
* * * *
* * * * *
```

PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

Java Ready

Ln 18, Col 2 Spaces: 4 UTF-8 CRLF 12:09 13-07-2024

17. Write a program to print the below pattern?

```
1
      1       1
        1       2       1
          1       3       3       1
            1       4       6       4       1
```

OUTPUT:

```
J pascaltriangle.java > pascaltriangle > main(String[])
 1  public class pascaltriangle {
 2      Run | Debug
 3      public static void main(String[] args) {
 4          int n=5;
 5          int num=1;
 6          int space=n;
 7          for(int i=0;i<n;i++){
 8              for(int k=1;k<space;k++){
 9                  System.out.print(" ");
10              }
11              num=1;
12              for(int j=0;j<=i;j++){
13                  System.out.print(num+" ");
14                  num=num*(i-j)/(j+1);
15              }
16              space--;
17          }
18      }
19  }
20
21
```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine. re-enable it, run 'Import-Module PSReadLine'.

```
PS C:\Users\kusha\OneDrive\Documents\java vs code> & 'C:\Program Files\IBM\SPSS\Statistics
ata\Roaming\Code - Insiders\User\workspacestorage\cce6980de64cf3261146ee044446caeab\redhat.j
angle'
 1
 1 1
 1 2 1
 1 3 3 1
 1 4 6 4 1
PS C:\Users\kusha\OneDrive\Documents\java vs code>
```

18. Write a program to print rectangle symbol pattern.

Get the symbol as input from user

OUTPUT:

The screenshot shows the Java VS Code interface. The code editor displays a Java program named `arrayrectangle.java` which prints a diamond shape based on user input. The terminal window below shows the program's output for 5 rows.

```
arrayrectangle.java | arrayrectangle.java | stringarangelettersalph | imberinarray.java | reverse.java | duplicateelements.java | arrayrectangle.java > Language Support for Java(TM) by Red Hat > arrayrectangle
1 import java.util.Scanner;
2 public class arrayrectangle {
3     public static void main(String[] args) {
4         Scanner a=new Scanner(System.in);
5         System.out.println("Enter Rows: ");
6         n=a.nextInt();
7         for(int i=0;i<n;i++){
8             for(int j=0;j<n-i;j++){
9                 System.out.print(" ");
10            }
11        }
12        System.out.println();
13    }
14 }
15 
```

TERMINAL

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\740cc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_0e18c00a\bin' 'arrayrectangle'
● Enter Rows:
5
* * * * *
* * * *
* *
* * * *
* * * * *
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> Ln 15, Col 2 Spaces: 4 UTF-8 CR
Windows Taskbar icons are visible at the bottom.
```

19. 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
```

OUTPUT:

```

final Help ← → ○ Java Vs Code
J reverse.java J duplicateelement J stringarrangelettersalph J addevenarray.java J perfectsquare.java J arraypattern.java X D ...
J arraypattern.java > Language Support for Java(TM) by Red Hat > arraypattern > main(String[])
1 public class arraypattern {
2     Run | Debug | Run main | Debug main
3     public static void main(String[] args) {
4         int n=3;
5         for(int i=0;i<n;i++){
6             for(int j=0;j<i;j++){
7                 System.out.print("1 ");
8             }
9             System.out.println();
10        }
11        for(int i=0;i<n;i++){
12            for(int j=i;j<n-1;j++){
13                System.out.print("1 ");
14            }
15        }
16    }
17 }

PROBLEMS 24 OUTPUT DEBUG CONSOLE TERMINAL PORTS
○ PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
● PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> c: cd 'c:\Users\jahna\OneDrive\Documents\Java Vs Code'; & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'c:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'arraypattern'
1
1 1
1 1 1
1 1
1
○ PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

Terminal: Ready
Ln 3, Col 17 Spaces: 4 UTF-8 CRLF ⓘ Java
12:14 13-07-2024 ⓘ

```

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

OUTPUT:

```

final Help ← → ○ Java Vs Code
J arrayinvertedfullpyramid.java X J stringarrangelettersalph J rectangle.java 1 J arraypascal.java J multiplicationtable.java 1 D ...
J arrayinvertedfullpyramid.java > Language Support for Java(TM) by Red Hat > arrayinvertedfullpyramid > main(String[])
1 public class arrayinvertedfullpyramid {
2     Run | Debug | Run main | Debug main
3     public static void main(String[] args){
4         int n=5;
5         for(int i=0; i<n; i++){
6             for(int k=0; k<i; k++){
7                 System.out.print(" ");
8             }
9             for(int j=0; j<n-1; j++){
10                System.out.print("* ");
11            }
12            System.out.println();
13        }
14    }
15 }

PROBLEMS 24 OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> 8 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'arrayinvertedfullpyramid'
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * *
* * *
* *
○ PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

Terminal: Ready
Ln 6, Col 36 Spaces: 4 UTF-8 CRLF ⓘ Java
12:14 13-07-2024 ⓘ

```

ASSIGNMENT-3

- 21.** Write a program to print the following pattern

Sample Input:

Enter the Character to be printed: %

Max Number of time printed: 3

```
%  
% %  
% % %
```

OUTPUT:

The screenshot shows the Java Vs Code interface. The left pane displays the code for a Java program named `arrayrighttriangle.java`. The code uses a nested loop to print a right-angled triangle of '%'. The right pane shows the terminal output where the user enters '3' and the program prints three lines of '%'. Below the terminal is a list of recent run configurations.

```
arrayrighttriangle.java 1 x Java Vs Code  
arrayrighttriangle.java > Language Support for Java(TM) by Red Hat > arrayrighttriangle > main(String[])  
1 import java.util.Scanner;  
2 public class arrayrighttriangle {  
3     public static void main(String[] args){  
4         int n;  
5         System.out.println("Enter No of Rows :");  
6         Scanner sc=new Scanner(System.in);  
7         n=sc.nextInt();  
8         for(int i=0;i<n;i++){  
9             for(int j=0;j<i+1){  
10                 System.out.print("% ");  
11             }  
12             System.out.println();  
13         }  
14     }  
15 }  
PROBLEMS 24 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> ^C  
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>  
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> cd "c:\Users\jahna\OneDrive\Documents\Java Vs Code"; & 'C:\Program Files\IBM\SPSS\Statistics\25>JRE\bin\java.exe' -cp 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\reduat.java' dt_vsJava Vs Code_8e18c80a\bin' 'arrayrighttriangle'  
Enter No of Rows :  
3  
%  
% %  
% % %  
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> |  
Ln 8, Col 24 Spaces: 4 UTF-8 CRLF 12:16 13-07-2024 Java
```

- 22.** Write a program to print hollow square symbol pattern?

OUTPUT:

The screenshot shows the Java Vs Code interface. The top bar has tabs for 'arrayhollowsquare.java', 'stringarrangelettersalph', 'java 1', 'arrayinvertedfullpyramid.java', and 'arrayhollowsquare.java'. The main editor window contains the following Java code:

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

The terminal below shows the output of the program:

```
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:/Users/jahna/AppData/Roaming/Code/User/workspaceStorage\749ecc9c81f4b75d684ae920d10202df/redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'arrayhollowsquare'
*****
*   *
*   *
*   *
*****
```

The status bar at the bottom indicates 'Ln 9, Col 37' and '12:16 13-07-2024'.

23. Write a program to print the below pattern

```
1
2 2
3 3 3
4 4 4 4
```

OUTPUT:

The screenshot shows the Java Vs Code interface. The code editor displays a Java file named `arrayrighttriangle.java` with the following content:

```
1 import java.util.Scanner;
2 public class arrayrighttriangle {
3     public static void main(String[] args){
4         int n;
5         System.out.print("Enter No of Rows :");
6         Scanner sc=new Scanner(System.in);
7         n=sc.nextInt();
8         for(int i=0;i<n;i++){
9             for(int j=0;j<i+1){
10                 System.out.print(i+" ");
11             }
12             System.out.println();
13         }
14     }
15 }
```

The terminal window below shows the execution of the program and its output:

```
% % %
○ ps C:\Users\jahna\OneDrive\Documents\Java Vs Code> ^C
○ ps C:\Users\jahna\OneDrive\Documents\Java Vs Code>
○ ps C:\Users\jahna\OneDrive\Documents\Java Vs Code> c:\cd 'c:\Users\jahna\OneDrive\Documents\Java Vs Code'; & 'C:\Program Files\IBM\SP
● ss\Statistics\25\JME\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae92d10202df\re
dhat.java\jdt_ws\Java vs code_8e18c80a\bin' 'arrayrighttriangle'
Enter No of Rows :
5

1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
○ PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
```

The status bar at the bottom indicates the current time is 12:17 on 13-07-2024.

24. Write a program to print the below pattern

```
1
4   9
16  25  36
49  64  81  100
```

OUTPUT:

```

arrayrighttriangle.java 1 x stringarrangelettersalph je.java frequency.java largestsubarray.java distinctchar D ...
J arrayrighttriangle.java > Language Support for Java(TM) by Red Hat > arrayrighttriangle > main(String[])
1 import java.util.Scanner;
2 public class arrayrighttriangle {
3     public static void main(String[] args){
4         int n;
5         System.out.println("Enter No of Rows :");
6         Scanner sc=new Scanner(System.in);
7         n=sc.nextInt();
8         int num=1;
9         for(int i=0;i<n;i++){
10             for(int j=0;j<i;j++){
11                 System.out.print((num*num)+" ");
12                 num++;
13             }
14             System.out.println();
15         }
16     }
17 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

4 4 4 4
5 5 5 5
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> ^
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> ^
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> c: cd 'c:/users/jahna/onedrive/documents/java vs code' & 'c:/program files/ibm/spss/statistics/25/bin/java.exe' '-cp' 'c:/users/jahna/appdata roaming/code/user/workspacestorage/749ecc9c81f4b75d664ae92d10202df/reduat.java' < jdt_ws>/java vs code_8e18c80a/bin' 'arrayrighttriangle'
Enter No Of Rows :
4
1
4 9
16 25 36
49 64 81 100
121 144 169 196 225
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> 

```

Java: Ready

25. Write a program to print the below pattern

```

1
2 2
3 3 3
4 4 4 4
3 3 3
2 2
1

```

OUTPUT:

```

arraypascal.java x multiplication.java stringarrangelettersalph vse.java string.java sumofarray.java ...
J arraypascal.java > Language Support for Java(TM) by Red Hat > arraypascal > main(String[])
1 public class arraypascal {
2     public static void main(String[] args){
3         int n=4;
4         for(int i=0;i<n;i++){
5             for(int k=0;k<i;k++){
6                 System.out.print((i)+" ");
7             }
8             System.out.println();
9         }
10        for(int i=n-1;i>=0;i--){
11            for(int j=1;j<=i;j++){
12                System.out.print((i)+" ");
13            }
14            System.out.println();
15        }
16    }
17 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

1
2 2
3 3 3
4 4 4 4
3 3 3
2 2
1
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> ^
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> ^
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> c: cd 'c:/users/jahna/onedrive/documents/java vs code' & 'c:/program files/ibm/spss/statistics/25/bin/java.exe' '-cp' 'c:/users/jahna/appdata roaming/code/user/workspacestorage/749ecc9c81f4b75d664ae92d10202df/reduat.java' < jdt_ws>/java vs code_8e18c80a/bin' 'arraypascal'

```

```

1
2 2
3 3 3
4 4 4 4
3 3 3
2 2
1
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> 

```

vse: Ready

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

OUTPUT:

The screenshot shows the Java Vs Code interface. The code editor displays the following Java code:

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

The terminal below shows the output of the program:

```
PS C:\Users\jaha\OneDrive\Documents\Java Vs Code> java arrayhollowsquare
$ $ $ $
$ $ $ $
$ $ $ $
$ $ $ $
```

The status bar at the bottom indicates "Java Ready".

27. Write a program to print inverted pyramid pattern.

Input: no of rows: 3

Output

```
*****
```

```
***
```

```
*
```

OUTPUT:

The screenshot shows the Java Vs Code interface. The code editor displays a Java file named `arrayinvertedfullpyramid.java` with the following content:

```

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

```

The terminal tab shows the following command-line session:

```

PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> cd "C:\Users\jahna\OneDrive\Documents\Java Vs Code" & 'C:\Program Files\IBM\SDP\statistics\25\JRE\bin\java.exe' -cp 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81fb75d604ae920d10202df\re
dhat\java\jdt_ws\Java Vs Code_be18c0ea\bin' 'arrayinvertedfullpyramid'
* *
* *
*
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

```

The status bar at the bottom right indicates the date and time: 13-07-2024 12:21.

General:

28. 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

OUTPUT:

```

final Help < > Java Vs Code
file reverse.java power3.java stringarrangelettersalph cationtable.java missingnumberinarray.java reverse.java ...
1 class reverse{
2     Run | Debug | Run main | Debug main
3     public static void main(String[] args) {
4         int n=123;
5         int rev=0;
6         while(n!=0){
7             int rem=n%10;
8             rev=rev*10 + rem ;
9             n=n/10;
10        }
11        System.out.println("Reversed number : ");
12        System.out.print(rev);
13    }
14 }

PROBLEMS 24 OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d60ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'reverse'
321
○ PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
○ PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
● PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d60ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'reverse'
Reversed number :
321
○ PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

Java Ready
Ln 10, Col 47 Spaces: 4 UTF-8 CRLF { Java 12:22 13-07-2024

```

29. 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

OUTPUT:

The screenshot shows the Java Vs Code interface. The terminal window at the bottom displays a list of run configurations:

- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> ^C
- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> cd 'c:\Users\jahna\OneDrive\Documents\Java Vs Code'; & 'c:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' -cp 'G:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\reduat.java\jdt_ws\Java Vs Code_be18c80a\bin' 'decimaltobinary'
- Result:
13
- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

The terminal status bar shows: Ln 9, Col 34 | Spaces: 4 | UTF-8 | CRLF | Java | 12:25 | 13-07-2024.

ASSIGNMENT-4

- 30.** 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:

6. 25
7. Eighteen
8. 12
9. -18
10. 34.5

OUTPUT:

The screenshot shows the Java Visual Studio Code interface. The code editor displays the following Java code:

```
generalvote.java // Language Support for Java(TM) by Red Hat > main(String[])
1 import java.util.Scanner;
2 public class generalvote {
3     public static void main(String[] args) {
4         int n;
5         Scanner sc=new Scanner(System.in);
6         System.out.println("Enter Age:");
7         n=sc.nextInt();
8         if(n>18){
9             System.out.println("You are Eligible to vote");
10        } else
11            System.out.println("You need extra "+(18-n)+" years to vote");
12    }
13
14
15 }
```

The terminal window shows the execution of the program:

```
PS C:\Users\jaha\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jaha\OneDrive\Documents\Java Vs Code\user\workspaceStorage\749ec9c81ffab75de04ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'generalvote'
Enter Age:
11
You need extra 7 years to vote
PS C:\Users\jaha\OneDrive\Documents\Java Vs Code>
```

The status bar at the bottom indicates "Java Ready".

- 31.** 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}

OUTPUT:

The screenshot shows the Java Visual Studio Code interface. The code editor displays the `gcdlcm.java` file with the following content:

```

1  public class gcdlcm {
2      Run | Debug | Run main | Debug main
3      public static void main(String[] args){
4          int n1=10;
5          int n2=20;
6          int a=n1,b=n2;
7          while(n1!=n2){
8              int r=n1%n2;
9              n1=n2;
10             n2=r;
11         }
12         System.out.println("GCD : "+n2);
13         System.out.println("LCM : "+((a*b)/n2));
14     }
}

```

The terminal below shows the execution of the program:

```

PS C:\Users\jahnna\OneDrive\Documents\Java Practice VS Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahnna\AppData\Roaming\Code\User\workspaceStorage\749ec5c81f4b75d68aae528d10282df\redhat\jdt_ls\java-project\bin' 'gcdlcm'
GCD : 10
LCM : 20

```

The status bar at the bottom indicates "Java: Ready".

- 32.** 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:

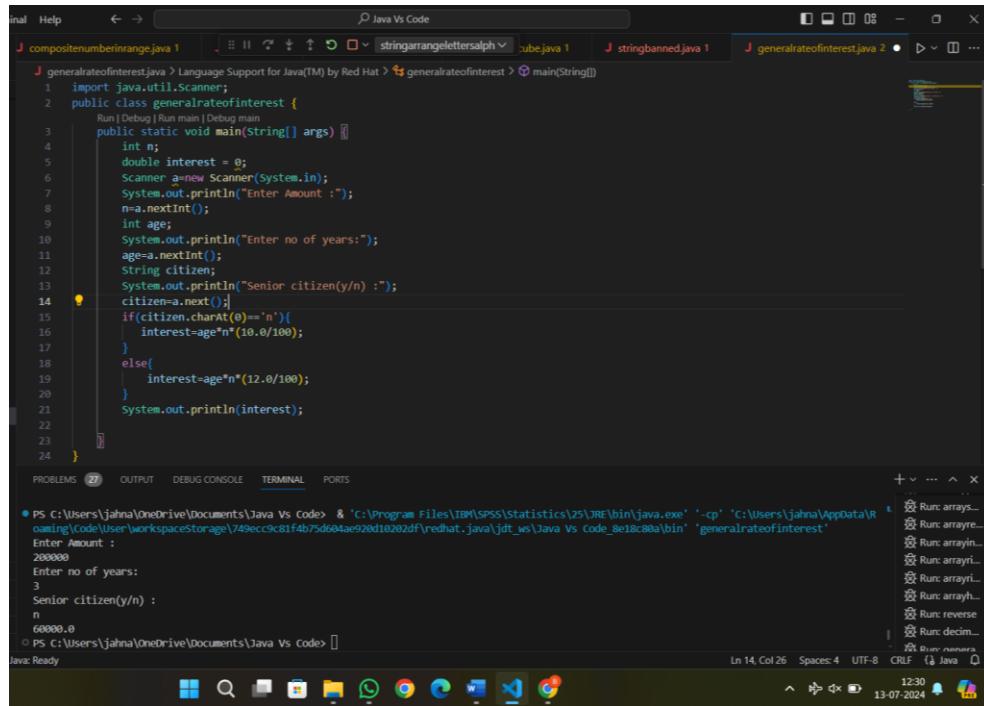
Interest: 60000

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

OUTPUT:



```

1  import java.util.Scanner;
2  public class generalrateofinterest {
3      Run | Debug | Run main | Debug main
4      public static void main(String[] args) {
5          int n;
6          double interest = 0;
7          Scanner a=new Scanner(System.in);
8          System.out.println("Enter Amount :");
9          n=a.nextInt();
10         int age;
11         System.out.println("Enter no of years:");
12         age=a.nextInt();
13         String citizen;
14         System.out.println("Senior citizen(y/n) :");
15         citizen=a.next();
16         if(citizen.charAt(0)=='n'){
17             interest=age*n*(10.0/100);
18         }
19         else{
20             interest=age*n*(12.0/100);
21         }
22         System.out.println(interest);
23     }
24 }
```

PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'c:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'c:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc3c81f4b75d64ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'generalrateofinterest'
Enter Amount :
20000
Enter no of years:
3
Senior citizen(y/n) :
n
60000.0

- 33.** Write a program to print the Fibonacci series.

Sample Input:

Enter the n value: 6

Sample Output:

0 1 1 2 3 5

OUTPUT:

```

1  import java.util.Scanner;
2  class print{
3      public static void main (String[] args){
4          int a=0;
5          int b=1;
6          int n;
7          int c;
8          Scanner sc=new Scanner(System.in);
9          System.out.println("Enter range :");
10         n=sc.nextInt();
11         System.out.print(a+" "+b+" ");
12         for (int i=2;i<n;i++){
13             c=a+b;
14             System.out.print(c+" ");
15             a=b;
16             b=c;
17         }
18     }
19 }

PROBLEMS 28 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> ^C
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
● PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> cd 'c:\Users\jahna\OneDrive\Documents\Java Vs Code'; & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'c:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81fb75d694ae920d10202df\redhat.java\jdt_ws\java Vs Code_8e18c80a\bin' 'print'
Enter range :
6
0 1 1 2 3 5 8
○ PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> ■

```

Java Ready

34. 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 = $0 + 1 + 3 + 8 + 21 = 33$)

CODE:

```
J generalfib.java > Language Support for Java(TM) by Red Hat > generalfib > main(String[])
1 import java.util.*;
2 public class generalfib {
3     Run | Debug | Run main | Debug main
4     public static void main(String[] args) {
5         int n1=0,n2=1;
6         int n;
7
8         Scanner a=new Scanner(System.in);
9         System.out.println("Enter Range: ");
10        n=a.nextInt();
11        ArrayList<Integer> arr=new ArrayList<>();
12
13        System.out.print("\nFibonacci : ");
14        System.out.print(n1+" "+n2+" ");
15        for(int i=1;i<(n*2);i++){
16            int c=n1+n2;
17            System.out.print(c+" ");
18            arr.add(c);
19            n1=n2;
20            n2=c;
21        }
22        int arrsum=0;
23        for(int i=0;i<arr.size();i++){
24            if(i%2==0){
25                arrsum=arrsum+arr.get(i);
26            }
27        }
28        System.out.print("\nArray sum : ");
29        System.out.print(arrsum);
30    }
31 }
32 }
```

OUTPUT:

```
Enter Range:
4

Fibonacci : 0 1 1 2 3 5 8 13 21
Array sum : 33
```

35. 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 = 50

OUTPUT:

The screenshot shows the Java Vs Code interface. The code editor displays a Java file named `generalmtontskip.java` with the following content:

```
1 import java.util.Scanner;
2 public class generalmtontskip {
3     public static void main(String[] args){
4         int m;
5         int n;
6         int k;
7         int l=0;
8         Scanner a=new Scanner(System.in);
9         System.out.println("Enter M value: ");
10        m=a.nextInt();
11        System.out.println("Enter N value: ");
12        n=a.nextInt();
13        System.out.println("Enter K value: ");
14        k=a.nextInt();
15        System.out.println("Values are: ");
16        int i=0;
17        for(i=m;i<n;i+=(k+1)){
18            System.out.print(i+" ");
19        }
20    }
21 }
22 }
```

The terminal window below the code editor shows the execution of the program:

```
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\CodeUser\workspaceStorage\749ecc9c81f4b75d604ae920d10282df\redhat-.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'generalmtontskip'
Enter M value:
59
Enter N value:
100
Enter K value:
7
Values are:
59 58 66 74 82 90 98
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
```

36. 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

CODE:

```

J compositenumberinrange.java > Language Support for Java(TM) by Red Hat > compositenumberinrange > compositenumberinrange.java
1 import java.util.Scanner;
2 public class compositenumberinrange {
3     Run | Debug | Run main | Debug main
4     public static void main(String[] args) {
5         int a;
6         int b;
7         Scanner sc=new Scanner(System.in);
8         System.out.println("Enter a value: ");
9         a=sc.nextInt();
10        System.out.println("Enter b value: ");
11        b=sc.nextInt();
12        System.out.println("Composite Number between " +a+ " and" +b);
13        for(int i=a;i<=b;i++){
14            if(compo(i)){
15                System.out.print(i+" ");
16            }
17        }
18    }
19    public static boolean compo(int num){
20        if(num<=3){
21            return false;
22        }
23        for(int i=2;i<=Math.sqrt(num);i++){
24            if(num%i==0){
25                return true;
26            }
27        }
28        return false;
29    }
}

```

OUTPUT:

```

Enter a value:
10
Enter b value:
20
Composite Number between10 and20
10 12 14 15 16 18 20

```

37. 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

OUTPUT:

The screenshot shows the Java Vs Code interface. The code editor displays a Java file named 'factorial.java' with the following content:

```
1 import java.util.Scanner;
2 class factorial {
3     public static void main(String[] args){
4         int fact=1;
5         Scanner a=new Scanner(System.in);
6         int n;
7         System.out.println("Enter number: ");
8         n=a.nextInt();
9         for(int i=1;i<=n;i++) {
10             fact=fact*i;
11         }
12         System.out.println("factorial of "+n+" is "+fact );
13     }
14 }
```

The terminal below shows the execution of the program:

```
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\statistics\25\RE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81fb75d94ae920d10202df\redhat.java\jdt_wt\Java Vs Code_8e18c88a\bin' 'factorial'
Enter number:
5
factorial of 5 is 120
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
```

38. 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

OUTPUT:

The screenshot shows the Java Vs Code interface. In the top bar, there are tabs for 'java', 'perfectnumber.java', 'leapyear', 'stringarrangelettersalph', 'generatelatinerest.java 2', 'generalfib.java 1', and 'leapyear.java 1'. The 'leapyear.java' tab is active, displaying the following Java code:

```

1 leapyear.java > Language Support for Java(TM) by Red Hat > leapyear > main(String[])
2 import java.util.Scanner;
3 public class leapyear {
4     public static void main(String[] args) {
5         Scanner a=new Scanner(System.in);
6         System.out.println("Enter Year : ");
7         n=a.nextInt();
8         if(n%4==0){
9             System.out.println("Leap Year");
10        }
11        else if(n%400==0 && n%100==0){
12            System.out.println("Leap Year");
13        }
14        else{
15            System.out.print("Not");
16        }
17    }
18 }
19

```

Below the code editor is the 'TERMINAL' tab, which shows the command-line output of the program's execution:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\dt_ws\Java Vs Code_8e18c80a\bin' 'leapyear'
Leap Year
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> <
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> <
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> < ; cd 'c:\Users\jahna\OneDrive\Documents\Java Vs Code'; & 'C:\Program Files\IBM\SP
SS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\re
dhat.java\dt_ws\Java Vs Code_8e18c80a\bin' 'leapyear'
Enter Year :
2400
Leap Year
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> ■

```

The terminal also shows the current working directory as 'C:\Users\jahna\OneDrive\Documents\Java Vs Code'.

39. 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

OUTPUT:

The screenshot shows the Java Vs Code interface. In the top bar, there are tabs for 'factors.java 1', 'perfectnumbe...', 'stringarrangelettersalph...', 'generatelinterestjava 2', 'generfib.java 1', and 'leapyear...'. Below the tabs, the code editor displays a Java program named 'factors.java' with the following content:

```
1 import java.util.Scanner;
2 public class factors {
3     public static void main(String[] args) {
4         Scanner s=new Scanner(System.in);
5         System.out.println("Enter number : ");
6         n=s.nextInt();
7         c=0;
8         for(int i=1;i<=n;i++){
9             if(n%i==0){
10                 c+=1;
11             }
12         }
13     }
14     System.out.println("No of Factors : ");
15     System.out.println(c);
16 }
17
18
19 }
```

Below the code editor is a terminal window titled 'TERMINAL'. It shows the command line output of running the program:

```
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> *c
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
● PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> c; cd 'c:\Users\jahna\OneDrive\Documents\Java Vs Code'; & 'c:\Program Files\IBM\SPSS\Statistics\25\REB\bin\java.exe' '-cp' 'c:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81fb75d604ae92d10202dF\re
dhat_jdt_ws\Java Vs Code_8e18c8ba\bin' 'factors'
Enter number :
100
No of Factors :
9
○ PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
```

The terminal also shows the current working directory as 'C:\Users\jahna\OneDrive\Documents\Java Vs Code'. At the bottom right of the terminal, there is a dropdown menu with several run configurations listed:

- Run: genera...
- Run: genera...
- Run: gdcm
- Run: genera...
- Run: genera...
- Run: print
- Run: genera...
- Run: compo...
- Run: factorial
- Run: leapyear
- Run: factors

40. 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

OUTPUT:

The screenshot shows the Java Vs Code IDE interface. The code editor displays a Java program named 'perfectnumber.java' which checks if a given number is perfect. The terminal below shows the execution of the program with input '6' and output 'It is perfect number'. A dropdown menu on the right lists various run configurations for other Java files in the workspace.

```
1 import java.util.Scanner;
2 public class perfectnumber {
3     public static void main(String[] args){
4         int n;
5         Scanner a=new Scanner(System.in);
6         System.out.println("Enter Number : ");
7         n=a.nextInt();
8         int sum=0;
9         for(int i=1;i<=n/2;i++){
10             if(n%i==0){
11                 sum+=i;
12             }
13         }
14         if(Sum==n){
15             System.out.println("it is perfect number");
16         }
17         else{
18             System.out.println("it is not perfect number");
19         }
20     }
21 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & "C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe" '-cp' 'c:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_Be18C00a\bin' 'perfectnumbe r'
Enter Number :
6
It is perfect number
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

+ ... ^

- Run: general...
- Run: gcd...
- Run: general...
- Run: general...
- Run: print...
- Run: general...
- Run: composition...
- Run: factorial...
- Run: leapyear...
- Run: factors...
- Run: perfect...

Ln 7, Col 23 Spaces:4 UTRF-8 CRLF 12:52 13-07-2024

41. 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

Test cases:

1. 12

2. 0

3. -0.5

4. 14.25

5. -296

OUTPUT:

The screenshot shows a Java code editor interface with a dark theme. In the code editor, there is a Java file named `squareandcube.java`. The code defines a class `squareandcube` with a `main` method. The code uses `Scanner` to read a double value from standard input, calculates its square and cube, and prints them to standard output. A yellow bug icon is visible next to the line `System.out.println("Enter a number : ");`. Below the code editor is a terminal window showing the command-line interface. The terminal shows the command to run the Java file, followed by the user input `0.6`, and the program's output showing the square and cube of 0.6.

```
J squareandcube.java > Language Support for Java(TM) by Red Hat > ⚙ squareandcube > ⏱ main(String[])

1 import java.util.Scanner;
2 public class squareandcube {
3     Run | Debug | Run main | Debug main
4     public static void main(String[] args) {
5         double n;
6         Scanner a=new Scanner(System.in);
7         System.out.println("Enter a number : ");
8         n=a.nextDouble();
9         double square=n*n;
10        double cube=n*n*n;
11        System.out.println("Square : "+square);
12        System.out.println("Cube : "+cube);
13    }
14
15

PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'squareandcube'
Enter a number :
0.6
Square : 0.36
Cube : 0.216
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
```

42. 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.

Test cases:

1. N = 0
2. N = -6
3. N = 2021
4. N = -14.5
5. N = -196

CODE:

```
J oddinrange.java > ...
1  public class oddinrange {
2      Run | Debug | Run main | Debug main
3      public static void main(String[] args) {
4          int n=7;
5          for(int i=0;i<n;i++){
6              if(i%2!=0){
7                  System.out.print(i+" ");
8              }
9          }
10     }
11 }
```

OUTPUT:

```
1 3 5
```

43. Program to find the frequency of each element in the array.

Sample Input & Output:

```
{1, 2, 8, 3, 2, 2, 2, 5, 1}
```

Pseudo:

Element | Frequency

Element	Frequency
1	2
2	4
8	1
3	1
4	1

CODE:

```

frequency.java > ...
1  public class frequency {
2      Run | Debug | Run main | Debug main
3      public static void main(String[] args) {
4          int[] arr={1,1,2,3,4,2,5,6,5,6};
5          int fre[]={};
6          int visited=-1;
7          for(int i=0;i<arr.length;i++){
8              int count=1;
9              for(int j=i+1;j<arr.length;j++){
10                  if(arr[i]==arr[j]){
11                      count++;
12                      fre[j]=visited;
13                  }
14                  if(fre[i]!=visited){
15                      fre[i]=count;
16                  }
17              }
18              System.out.println("Element | frequency");
19              for(int i=0;i<fre.length;i++){
20
21                  if(fre[i]!=visited){
22                      System.out.println(arr[i] + " | " + fre[i]);
23                  }
24              }
25          }
26      }
27  }

```

OUTPUT:

Element	frequency
1	2
2	2
3	1
4	1
5	2
6	2

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

Sample Input:

Enter number: 153

Sample Output:

Given number is Armstrong number

Test cases:

1. 370
2. 1
3. 371
4. 145678
5. 0.21345

CODE:

```
J armstrong.java > Language Support for Java(TM) by Red Hat > 🛡 armstrong > ⚒ main(String[])
1  public class armstrong {
2      Run | Debug | Run main | Debug main
3      public static void main(String[] args) {
4          int res=0;
5          int rem,n=153;
6          int temp=n,digit=0;
7          while (temp>0){
8              temp=temp/10;
9              digit++;
10         }
11         temp=n;
12         while (n!=0){
13             rem=n%10;
14             res+= $(Math.pow(rem,digit))$ ;
15             n=n/10;
16         }
17         if (res==temp){
18             System.out.println(temp+" Armstrong");
19         }
20         else{
21             System.out.println(temp+" Not");
22         }
23     }
24 }
```

PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'armstrong'
153 Armstrong
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
```

45. 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

Test cases:

1. N = 2, 158
2. N = 3, 14
3. N = 4, 0148
4. N = 1, 0004
5. N = 4, 7263

CODE:

```

J sumofdigtsingledigit.java > Language Support for Java(TM) by Red Hat > sumofdigtsingledigit > main(String[])
1 import java.util.Scanner;
2 class sumofdigtsingledigit{
3     Run | Debug | Run main | Debug main
4     public static void main(String[] args){
5         int n;
6         int val;
7         Scanner a=new Scanner(System.in);
8         System.out.println("Enter N value : ");
9         n=a.nextInt();
10        System.out.println("Enter "+n+" Digit Number :");
11        val=a.nextInt();
12        int sum=0;
13        while(val>0||sum>9){
14            if(val==0){
15                val=sum;
16                sum=0;
17            }
18            int rem=val%10;
19            sum=sum+rem;
20            val=val/10;
21        }
22        System.out.println("Sum is :" +sum);
23    }
}

PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'sumofdigtsingledigit'
Enter N value :
2
Enter 2 Digit Number :
18
Sum is :9
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

```

46. 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

Test cases:

1. 1225
2. 9801
3. 1827
4. -100
5. 0

OUTPUT:

The screenshot shows a Java code editor interface. At the top, there's a breadcrumb navigation bar: J perfectsquare.java > Language Support for Java(TM) by Red Hat > perfectsquare > main(String[]). Below this is the code editor area with the following Java code:

```
1 public class perfectsquare {
2     Run | Debug | Run main | Debug main
2     public static void main(String[] args){
3         int n=6561;
4         double sqr=Math.sqrt(n);
5         System.out.println(sqr+(", -"+sqr));
6     }
7 }
```

Below the code editor is a toolbar with tabs: PROBLEMS (35), OUTPUT, DEBUG CONSOLE, TERMINAL (selected), and PORTS.

The TERMINAL tab shows the following output from the terminal window:

```
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'perfectsquare'
● 81.0, -81.0
○ PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
```

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

Input =27

Output= true

Explanation: $27=3^3$

Test cases:

1. 12
2. abc@45
3. 1827
4. -100
5. 0

OUTPUT:

```
J power3.java > Language Support for Java(TM) by Red Hat > power3 > main(String[])
1 import java.util.Scanner;
2 public class power3 {
3     Run | Debug | Run main | Debug main
4     public static void main(String[] args) {
5         int n;
6         int found=0;
7         Scanner a=new Scanner(System.in);
8         System.out.println("Enter to check cube: ");
9         n=a.nextInt();
10        for(int i=0;Math.pow(3,i)<=n;i++){
11            if(Math.pow(3,i)==n){
12                found=1;
13            }
14        }
15        if(found==1){
16            System.out.println("True");
17        }
18        else{
19            System.out.println("False");
20        }
21    }
22
23 }
```

PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'power3'
Enter to check cube:
27
True
○ PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

48. 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]

Explanation: After calling your function, the input array is modified to [1, 0, 0, 2, 3, 0, 0, 4]

OUTPUT:

The screenshot shows a Java code editor with the following code:

```
1 public class zerooccurrences {
2     Run | Debug | Run main | Debug main
3     public static void main(String[] args) {
4         int arr[]={1,2,0,5,0,6};
5         int i=0;
6         while(i<arr.length-1){
7             if(arr[i]==0){
8                 for(int j=arr.length-1;j>=i+2;j--){
9                     arr[j]=arr[j-1];
10                }
11                arr[i+1]=0;
12                i=i+2;
13            } else{
14                i+=1;
15            }
16        }
17        for(int j=0;j<arr.length;j++){
18            System.out.print(arr[j]+" ");
19        }
20    }
21 }
22 }
```

The code uses a two-pass approach. The first pass iterates from index 0 to n-2, moving zeros to their correct positions (index i+2). The second pass prints the array elements.

Below the code editor is a terminal window showing the execution of the program:

```
PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS +
```

```
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> ^C
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> c:; cd 'c:\Users\jahna\OneDrive\Documents\Java Vs Code'; & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'c:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\re
dhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'zerooccurrences'
● 1 2 0 0 5 0
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
```

49. 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

Explanation: n = 3 since there are 3 numbers, so all numbers are in the range [0, 3]. 2 is the missing number in the range since it does not appear in nums.

OUTPUT:

The screenshot shows a Java code editor with a dark theme. The code is a Java program named `missingnumberinarray.java` that finds the missing number in an array. The code uses a formula to calculate the sum of numbers from 1 to n, then subtracts the sum of elements in the array to find the missing number. A yellow warning icon is visible near the end of the code. Below the code editor is a terminal window showing the command run and the output "Missing Number is 2".

```
J missingnumberinarray.java > Language Support for Java(TM) by Red Hat > missingnumberinarray > main(String[])
1  public class missingnumberinarray {
2      Run | Debug | Run main | Debug main
3      public static void main(String[] args) {
4          int[] arr={1,0,3};
5          int n=arr.length;
6          int Sum=0;
7          int formula=(n*(n+1))/2;
8          for(int i=0;i<n;i++){
9              Sum+=arr[i];
10         }
11         int miss=formula-Sum;
12         System.out.println("Missing Number is "+miss);
13     }
14
15
```

PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'missingnumberinarray'
Missing Number is 2
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
```

50. 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

Explanation: The subarray [4, -1, 2, 1] has the largest sum 6.

OUTPUT:

```
J largestsubarray.java > Language Support for Java(TM) by Red Hat > largestsubarray > main(String[])
1  public class largestsubarray {
2      Run | Debug | Run main | Debug main
3      public static void main(String[] args){
4          int arr[]={-2,1,-3,4,-1,2,1,-5,4};
5          int sum=0;
6          int max=arr[0];
7          for(int i=0;i<arr.length;i++){
8              sum+=arr[i];
9              if(sum>max) max=sum;
10             if(sum<0) sum=0;
11         }
12         System.out.println("Largest Subarray is "+max);
13     }
14 }
15
```

PROBLEMS 35 OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'largestsubarray'
 Largest Subarray is 6
- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>

51. Write a program to print the multiplication table of number m up to n.

Sample Input:

M = 4

N = 5

Sample Output:

1x4=4

2x4=8

3x4=12

4x4=16

5x4=20

Test cases:

M = 6, N = -3

M = -3, N = 5

M = 4, N = 0

M = 0, N = 0

M = -5, N = -5

OUTPUT:

```
J multiplicationtable.java > Language Support for Java(TM) by Red Hat > multiplicationtable > main(String[])
1 import java.util.Scanner;
2 public class multiplicationtable {
3     Run | Debug | Run main | Debug main
4     public static void main(String[] args){
5         int n;
6         Scanner a=new Scanner(System.in);
7         System.out.println("Enter the Number you need multiplication Table : ");
8         n=a.nextInt();
9         for(int i=1;i<=10;i++){
10             System.out.println(n+"*"+i+"="+n*i);
11         }
12     }

```

PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
5*1=5
5*2=10
5*3=15
5*4=20
5*5=25
5*6=30
5*7=35
5*8=40
5*9=45
5*10=50
```

PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>