

CSA0987 – JAVA PROGRAMMING\\ ASSIGNMENT – 1

NAME: O. Bhanu Prakash Reddy
Reg No:192372075

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 a Java code editor with several tabs at the top: reversestring.java (active), countvowels.java, printvowelsandconsonants.java, and stringchars. The reversestring.java tab contains the following code:

```
1 import java.util.Scanner;
2 public class reversestring {
3     public static void main(String[] args) {
4         String original,rev="";
5         System.out.println("Enter String:");
6         Scanner in=new Scanner(System.in);
7         original=in.nextLine();
8         for(int i=original.length()-1;i>=0;i--){
9             rev=rev+original.charAt(i);
10        }
11        System.out.println("Reversed String : "+rev);
12    }
13 }
```

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

```
PROBLEMS 25 OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\18.0\bin\java.exe' -jar 'C:\Program Files\IBM\SPSS\Statistics\18.0\bin\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:

Output 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:

```
validuser.java 1 J squareando... || ⌂ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ stringarrangelettersalph... iterest.java 2 J generalfib.java 1 J leapyear.java 1 J arraypr... ▾
```

```
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         public static void main(String[] args) {
5             String user;
6             Scanner a=new Scanner(System.in);
7             System.out.println("Enter Username : ");
8             user=a.nextLine();
9             String res="^([a-zA-Z0-9_]{0,17}$)";
10            if(user.matches(res))
11                System.out.println("Valid Username");
12            else
13                System.out.println("Not valid");
14 }
```

The terminal below shows the execution of the code:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... x
```

```
PS C:\Users\jahn... & 'c:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahn...\\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae92d10202df\redhat.java.jdt_ws\Java vs Code_8e18c88a\bin' 'validuser'
Enter Username :
balu456
Valid Username
PS C:\Users\jahn...\\Documents\Java Vs code |
```

The sidebar on the right lists various run configurations:

- Run: general
- Run: genera...
- Run: genera...
- Run: genera...
- Run: compo...
- Run: factors
- Run: leapyear
- Run: perfect
- Run: factorial
- 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

Jack

Order(A/D) : A

Sample Output:

Apple

Banana

Carrot

Jack

Radish

OUTPUT:

The screenshot shows a Java code editor and a terminal window. The code in the editor is a Java program named 'sortlistalphabet.java' that sorts three names in ascending order. The terminal window shows the execution of the program and its 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 }
```

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' -jar AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs abet
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:

The screenshot shows the VS Code interface with the following details:

- Editor:** A Java code editor window titled "special.java". The code prints each character of the string "hello @456%" if it is neither a letter nor a digit. The code is as follows:

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

- Terminal:** The terminal tab is active, showing the command line history:
 - 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>
- Bottom Navigation:** PROBLEMS (35), OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), PORTS.

- 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:

The screenshot shows the Visual Studio Code interface. The top part displays a Java code editor with the following code:

```
1 import java.util.Scanner;
2 public class countvowels {
3     public static void main(String[] args) {
4         String original;
5         int count=0;
6         System.out.print("Enter String ");
7         Scanner in=new Scanner(System.in);
8         original=in.nextLine();
9         original=original.toLowerCase();
10        int len=original.length();
11        for(int i=0;i<len;i++){
12            if(original.charAt(i)=='a'||original.charAt(i)=='e'||original.charAt(i)=='i'||original.charAt(i)=='o'||original.charAt(i)=='u'){
13                count+=1;
14            }
15        }
16        System.out.println("No of Vowels :"+count);
17    }
18 }
```

The bottom part shows the 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\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>
```

The right sidebar shows the file explorer with several run configurations:

- power (Run: r)
- Run: r
- Run: s
- Run: s
- Run: c

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            }
19            else if(Character.isLetter(ch)){
20                Con+=ch+" ";
21            }
22        }
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\statistics\25\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 stringrangelettersalph {
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' 'stringrangelettersalph'
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

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 1, printvow..., strin..., stringarrangelettersalph..., and stringremovevowels.java 1. The code for stringarrangelettersalph... is displayed:

```

1 public class stringarrangelettersalphaverse {
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 }

```

The terminal below shows the execution of the program:

```

PS C:\Users\jahnalOneDrive\Documents\Java Vs Code> ^C
PS C:\Users\jahnalOneDrive\Documents\Java Vs Code> PS C:\Users\jahnalOneDrive\Documents\Java Vs Code> c:; cd 'c:\Users\jahnalOneDrive\Documents\Java Vs Code' & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahnalAppData\Roaming\Code\User\workspaceStorage\749ecc9c81fb75d604ae920d10202df\redhat.java\dt_ws\Java Vs Code_8e18c80a\bin' 'stringarrangelettersalphaverse'
Enter String :
MOSQUE
Alphabetical Order : USQOME
PS C:\Users\jahnalOneDrive\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 1, printvow..., strin..., stringarrangelettersalph..., and stringremovevowels.java 1. The code for stringremovevowels.java is displayed:

```

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 }

```

The terminal below shows the execution of the program:

```

PS C:\Users\jahnalOneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahnal\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\jahnalOneDrive\Documents\Java Vs Code>

```

ASSIGNMENT-2

11. Write a program for matrix multiplication?

Sample Input:

Mat1 = $\begin{matrix} 1 & 2 \\ 5 & 3 \end{matrix}$

Mat2 = $\begin{matrix} 2 & 3 \\ 4 & 1 \end{matrix}$

Sample Output:

Mat Sum = $\begin{matrix} 10 & 5 \\ 22 & 18 \end{matrix}$

OUTPUT:

The screenshot shows the Java Visual Studio Code interface. The code editor has tabs for arraysmatrixmul.java, printvowelsandconsonants.java, stringarrangelettersalph.java, verse.java, and stringremovevowels.java. The arraysmatrixmul.java tab is active, displaying the following Java code:

```
public class arraysmatrixmul {
    public static void main(String[] args) {
        int[][] mat1={{1,2,3},{3,4,5},{1,1,1}};
        int[][] mat2={{1,1,1},{1,1,1},{1,1,1}};
        int[][] result=new int[mat1.length][mat2[0].length];
        for(int i=0;i<mat1.length;i++){
            for(int j=0;j<mat1[0].length;j++){
                for(int k=0;k<mat2[0].length;k++){
                    result[i][j]+=mat1[i][k]*mat2[k][j];
                }
            }
        }
        for(int i=0;i<mat1.length;i++){
            for(int j=0;j<mat2[0].length;j++){
                System.out.print(result[i][j]+" ");
            }
            System.out.println();
        }
    }
}
```

The terminal below shows the execution of the program 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\749ecc9c81f4b75d604ae920d10202df\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>
```

The status bar at the bottom indicates the code is ready.

12. Write a program for matrix addition?

Sample Input:

Mat1 = $\begin{matrix} 1 & 2 \\ 5 & 3 \end{matrix}$

Mat2 = $\begin{matrix} 2 & 3 \\ 4 & 1 \end{matrix}$

Sample Output:

Mat Sum = $\begin{matrix} 3 & 5 \\ 9 & 4 \end{matrix}$

OUTPUT:

The screenshot shows the Java Vs Code interface. The terminal window displays the output of running the 'arraysmatrixadd.java' program, which adds two 3x3 matrices and prints the result.

```

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\749ecc9c81f4b75d604ae92d10202df\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>

```

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:

The screenshot shows the Java Vs Code interface. The terminal window displays the output of running the 'arraymergesorted.java' program, which merges two sorted arrays (arr1 and arr2) into arr3.

```

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\749ecc9c81f4b75d604ae92d10202df\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>

```

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 displays 'arraycomposite.java' with the following content:

```
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;
    }
}
```

The terminal tab shows the command run 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/worksheets/storage/49ecc9c81f4b75dc04ae920d10202df/redhat.java/jdt_ws/Java Vs Code_0e18c80a/bin' 'arraycomposite'
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
```

The status bar at the bottom indicates: Ln 8, Col 14 Spaces:4 UTF-8 CRLF 1208 13-07-2024

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 displays a Java file named `arraypyramid.java` with the following content:

```

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<=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 }

```

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

```

PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL PORTS
/home/jahna/Code/User/workspaceStorage/749ecc9c81f4b75d604ae920d10202df/redhat.java:jdt_ws\Java_Vs_Code_Be18c80a\bin\arraypyramid
Enter Rows:
5
*
**
* *
* ***
* ****
* *****

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

```

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

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              System.out.println();
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\cce6980de64cf3261146ee04446cae3\redhat.java'
 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 file named `arrayrectangle.java` with the following content:

```
1 import java.util.Scanner;
2 public class arrayrectangle {
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;j++){
10                 System.out.print("* ");
11             }
12         }
13     }
14 }
```

The terminal window below shows the execution of the program 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\740cc9c81f4b75d604ae920d10202df\redhat.java\jdt_ws\Java Vs Code_0e18c00a\bin' 'arrayrectangle'
● Enter Rows:
5
*****
* * *
* * *
* * *
* * *
* * *

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

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:

The screenshot shows the Java Vs Code interface. The top bar has tabs for 'File', 'Help', and 'Java Vs Code'. The title bar shows multiple open files: 'reverse.java', 'duplicateelen.java', 'stringarrangelettersalph.java', 'addevenarray.java', 'perfectsquare.java', and 'arraypattern.java'. The main area displays the code for 'arraypattern.java'.

```
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;j++){
13                System.out.print("1 ");
14            }
15            System.out.println();
16        }
17    }
18 }
```

The bottom navigation bar includes 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active, showing the following command history:

- 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\7a9ecc9c81fb7sd6ae92d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c88a\bin' 'arraypattern'

The terminal output shows the execution of the program:

```
1
1 1
1 1 1
1 1
```

The status bar at the bottom right shows 'Ln 3, Col 17' and 'UTF-8'.

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

OUTPUT:

The screenshot shows the Java Vs Code IDE interface. The code editor displays a Java file named `arrayinvertedfullpyramid.java`. The terminal window shows the execution of the program, which prints an inverted pyramid of asterisks. The status bar at the bottom indicates the code is ready.

```
arrayinvertedfullpyramid.java
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-i;j++){
10                 System.out.print("* ");
11             }
12             System.out.println();
13         }
14     }
15 }
```

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\49ecc9c81f4b75d60ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'arrayinvertedfullpyramid'

* *
* *
* *
* *
*

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

Java: Ready

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 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.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 }
```

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

```
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> arrayrighttriangle  
Enter No of Rows :  
3  
% % %
```

The output window shows the printed pattern:

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

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

OUTPUT:

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

```
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 running 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\749ecc9c81fb75d6daef920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'arrayhollowsquare'
* * * *
*   *
*   *
*   *
* * * *
```

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

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 top bar has tabs for 'arrayrighttriangle.java' (active), 'atle.java', 'stringarrangelettersalphaje.java', 'frequency.java', 'largestsubarray.java', and 'distinctchar.java'. Below the tabs is the code editor with the following Java code:

```
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;j++){
10                 System.out.print(i+ " ");
11             }
12             System.out.println();
13         }
14     }
15 }
```

The code is highlighted with syntax coloring. A yellow lightbulb icon is on line 10, indicating a warning or suggestion. The bottom part of the interface shows the terminal 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\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d694ae92d10202df\redundant_code_8e18c00a\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 terminal also shows the Java taskbar at the bottom.

24. Write a program to print the below pattern

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

OUTPUT:

The screenshot shows the Java Vs Code IDE interface. The code editor displays a Java program named `arrayrighttriangle.java`. The terminal window shows the output of running the program, which prints a right-angled triangle pattern of numbers. The file list at the top shows several other Java files like `stringarrangelettersalph.java`, `je.java`, `frequency.java`, `largestsubarray.java`, and `distinctchar.java`.

```
arrayrighttriangle.java 1 x stringarrangelettersalph.java frequency.java largestsubarray.java distinctchar.java ...
```

```
import java.util.Scanner;
public class arrayrighttriangle {
    public static void main(String[] args){
        int n;
        System.out.println("Enter No of Rows :");
        Scanner sc=new Scanner(System.in);
        n=sc.nextInt();
        int num=1;
        for(int i=0;i<n;i++){
            for(int j=0;j<i;j++){
                System.out.print((num*num)+" ");
                num++;
            }
            System.out.println();
        }
    }
}
```

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\jREbin\java.exe' '-cp' 'C:\Users\jahna\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d604ae92ad10202df\reduat\java\jdt_ws\Java Vs Code_0e18c00a\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:

The screenshot shows the Java Vs Code interface with the following code in the editor:

```

public class arraypascal {
    public static void main(String[] args){
        int n=4;
        for(int i=0;i<n;i++){
            for(int k=0;k<i;k++){
                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();
        }
    }
}

```

The terminal window shows the output of the program:

```

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

```

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

OUTPUT:

The screenshot shows the Java Vs Code interface with the following code in the editor:

```

public class arrayhollowsquare {
    public static void main(String[] args) {
        int n=4;
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                if(i==0||i==n||j==0||j==n-1)
                    System.out.print("$ ");
                else
                    System.out.print("  ");
            }
            System.out.println();
        }
    }
}

```

The terminal window shows the output of the program:

```

$ $ $ $
$ $ $ 
$ $ $ 
$ $ $ 
$ $ $ $

```

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        }
12    }
13 }
```

The terminal window below shows the execution of the program:

```
PS C:\Users\jahnna\OneDrive\Documents\Java Vs Code> c; cd '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\workspaces\storage\749ecc9c81f4b75d604ae920d10202df\re dir\Java\jdt_ws\Java vs Code_be18c80a\bin' 'arrayinvertedfullpyramid'
*
*
*
PS C:\Users\jahnna\OneDrive\Documents\Java Vs Code>
```

The status bar at the bottom indicates the current file is `arrayinvertedfullpyramid.java`.

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:

```

1  class reverse{
2     public static void main(String[] args) {
3         int n=123;
4         int rev=0;
5         while(n!=0){
6             int rem=n%10;
7             rev=rev*10 + rem ;
8             n=n/10;
9         }
10        System.out.println("Reversed number : ");
11        System.out.print(rev);
12    }
13 }

```

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\749ecc9c81fb75d504ae920d10202dF\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 & 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\749ecc9c81fb75d504ae920d10202dF\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'reverse' Reversed number : 13
- PS C:\Users\jahna\OneDrive\Documents\Java Vs Code

Java Ready Ln 10, Col 47 Spaces: 4 UTF-8 CRLF 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:

$(1011)_2 = (13)_{10}$.

Test cases:

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

OUTPUT:

The screenshot shows the Java Vs Code interface. The top bar has tabs for 'decimaltobinary.java', 'odd', 'stringarrangelettersalph...', 'distinctcharactercount.java', and 'sumofdigitssingledigit.java'. The main editor window displays the following Java code:

```
1 public class decimaltobinary {
2     Run | Debug | Run main | Debug main
3     public static void main(String[] args) {
4         int n=11;
5         String bin=Integer.toBinaryString(n);
6         StringBuilder string=new StringBuilder();
7         string.append(bin);
8         string.reverse();
9         int deci=Integer.parseInt(string.toString(), 2);
10        System.out.println("Result: ");
11        System.out.println(deci);
12    }
13 }
14
15
```

The terminal below shows the command to run the file and its output:

```
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> java decimaltobinary
Result:
13
```

The terminal also lists several recent run configurations:

- Run: arrays...
- Run: arrayp...
- Run: arrayp...
- Run: arrayp...
- Run: arrays...
- Run: arrays...
- Run: arrayin...
- Run: arrayin...
- Run: arrayi...
- Run: arrayh...
- Run: reverse
- Run: decim...

The status bar at the bottom indicates 'Ln 9, Col 34' and '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 Vs Code interface. The code editor displays the `generalvote.java` file with the following content:

```
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 below shows the execution of the program:

```
PS C:\Users\jaina\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:/Users/jaina/AppData/Roaming/Code/User/workspaceStorage/749ecc9c81fab75d604ae92ed10202df/redhat:java/jdt_ws/Java Vs Code_8e18c88a/bin' 'generalvote'
Enter Age:
11
You need extra 7 years to vote
PS C:\Users\jaina\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 VS 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 window below shows the execution of the program:

```

PS C:\Users\jahnava\OneDrive\Documents\Java Practice VS Code> & "C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe" ".cp" 'C:\Users\jahnava\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81f4b75d04ae920d10202df\redhat.java\jdt_ws\jdt.ls\java-project\b\gcdlcm'
GCD : 10
LCM : 20

```

The status bar at the bottom indicates the terminal is 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

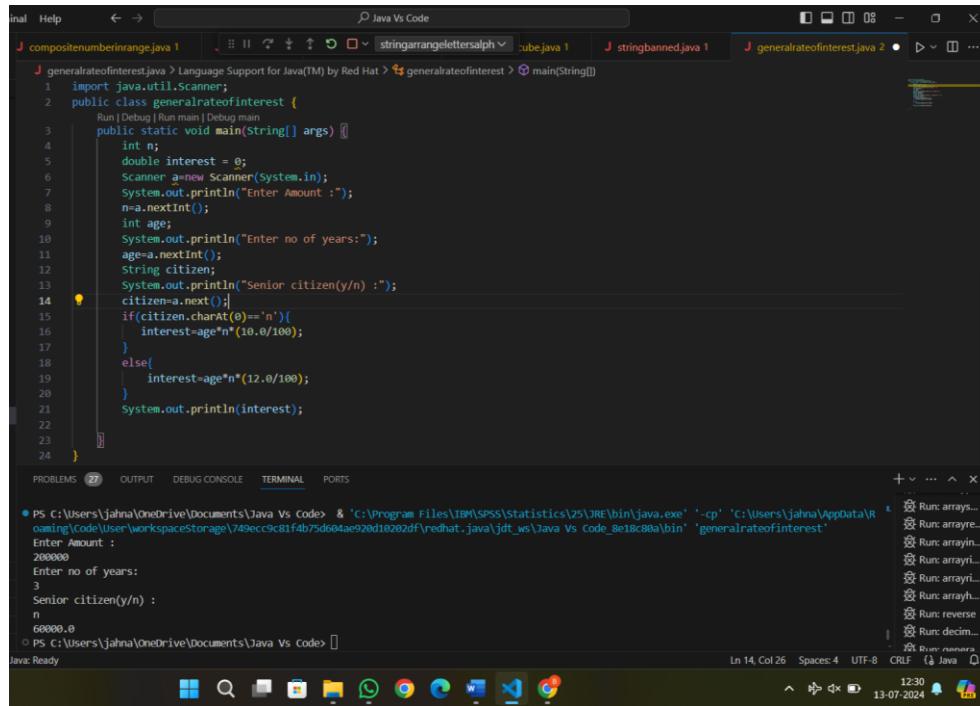
Test Cases:

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:



```
Java Vs Code
generalrateofinterest.java 1 stringarrangelettersalph... cube.java 1 stringbanned.java 1 generalrateofinterest.java 2
generalrateofinterest > Language Support for Java(TM) by Red Hat > generalrateofinterest > main(String[])
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        age=a.nextInt();
12        String citizen;
13        System.out.println("Senior citizen(y/n) :");
14        citizen=a.next();
15        if(citizen.charAt(0)=='n'){
16            interest=age*n*(10.0/100);
17        }
18        else{
19            interest=age*n*(12.0/100);
20        }
21        System.out.println(interest);
22    }
23 }
24
PROBLEMS 27 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\749ecc9c81fb75d64ae920df\redhat_java\jdk_ws\Java Vs Code_8e18c80a\bin' 'generalrateofinterest'
Enter Amount :
200000
Enter no of years:
3
Senior citizen(y/n) :
n
60000.0
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
Ln 14, Col 26 Spaces: 4 UTF-8 CRLF Java
12:30 13-07-2024
```

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.print("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\jahnajahn\OneDrive\Documents\Java Vs Code> ^c
PS C:\Users\jahnajahn\OneDrive\Documents\Java Vs Code> PS C:\Users\jahnajahn\OneDrive\Documents\Java Vs Code> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahnajahn\OneDrive\Documents\Java Vs Code' & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\jahnajahn\AppData\Roaming\Code\User\workspaceStorage\749ecc9c81fdb75d694ae920d10202df\redhat\java\jdt_wls\java Vs Code_8e18c80a\bin' 'print'
Enter range :
6
0 1 1 2 3 5 8
PS C:\Users\jahnajahn\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}

```

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 IDE interface. The code editor displays a Java program named `generalmtskip.java`. The terminal below shows the execution of the program, prompting for values M, N, and K, and then printing the values 50, 58, 66, 74, 82, 90, 98. The status bar at the bottom right indicates the current date and time as 13-07-2024.

```
J generalmtskip.java 4 x J com 11:11 stringrangeletteralph 1 ectnumber.java J squareandcube.java 1 J stringbanned.java 1

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

PROBLEMS 0 OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... x
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\749ecc9c81f4b75d64ae920d10202df\redhat.java\jdt_ws\Java Vs Code_8e18c80a\bin' 'generalmtskip'
Enter M value:
50
Enter N value:
100
Enter K value:
7
Values are:
50 58 66 74 82 90 98
PS C:\Users\jahnna\OneDrive\Documents\Java Vs Code> |
```

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

Sample Input:

$$\bar{A} = 12$$

$$B = 19$$

Sample Output

14, 15,

- st 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    }
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         Scanner a=new Scanner(system.in);
5         int n;
6         System.out.println("Enter number: ");
7         n=a.nextInt();
8         for(int i=1;i<=n;i++) {
9             fact=fact*i;
10        }
11        System.out.println("factorial of "+n+" is "+fact );
12    }
13 }
14 }
```

The terminal window below shows the execution of the program:

```
PS 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\749ecc9c81f4b75d694ae920d10202df\redhat.java\java Vs Code_8e18c80a\bin' 'factorial'
Enter number:
5
factorial of 5 is 120
PS C:\Users\jahna\OneDrive\Documents\Java vs Code>
```

The status bar at the bottom right indicates the date and time: 13-07-2024 12:46.

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:

```

1  leapyear.java > Language Support for Java(TM) by Red Hat > leapyear > main(String[])
2  public class leapyear {
3      public static void main(String[] args) {
4          Scanner a=new Scanner(System.in);
5          System.out.println("Enter Year : ");
6          n=a.nextInt();
7          if(n%4==0){
8              System.out.println("Leap Year");
9          }
10         else if(n%400==0 && n%100==0){
11             System.out.println("Leap Year");
12         }
13         else{
14             System.out.print("Not");
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' '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\jdt_ws\Java Vs Code_8e18c80a\bin' 'leapyear'
Enter Year :
2000
Leap Year
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> ■

```

ava: Ready

Ln 7, Col 20 Spaces: 3 UTF-8 CRLF 1248 13-07-2024

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 Microsoft Visual Studio Code interface. The left pane displays a Java file named 'factors.java' with the following code:

```
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         int n=s.nextInt();
7         int c=0;
8         for(int i=1;i<=n;i++){
9             if(n%1==0){
10                 c+=1;
11             }
12         }
13     }
14     System.out.println("No of Factors : ");
15     System.out.println(c);
16 }
17
18
19
```

The right pane shows a terminal window with the following session:

```
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code> cd "C:\Users\jahna\OneDrive\Documents\Java Vs Code"
PS 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\749ecc9c81f4b75d504ae92d10202df\rehat\Java\jdt_ws\Java Vs Code_8e18c80a\bin" "factors"
Enter number :
100
No of Factors :
9
PS C:\Users\jahna\OneDrive\Documents\Java Vs Code>
```

The status bar at the bottom indicates the terminal is in line 14, column 41, with 4 spaces and UTF-8 encoding.

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 file named `perfectnumber.java` with the following content:

```
1 import java.util.Scanner;
2 public class perfectnumber {
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 Number : ");
8         n=a.nextInt();
9         int sum=0;
10        for(int i=1;i<=n/2;i++){
11            if(n%i==0){
12                sum+=i;
13            }
14        }
15        if(sum==n){
16            System.out.println("It is perfect number");
17        }
18        else{
19            System.out.println("It is not perfect number");
20        }
21    }
22 }
```

The terminal below the code editor shows the output of running the program with the input `6`, which outputs `It is perfect number`. A dropdown menu on the right side of the terminal shows various run configurations for other Java files in the workspace.

ASSIGNMENT-5

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

Sample Input:

Given Number: 0.6

Sample Output:

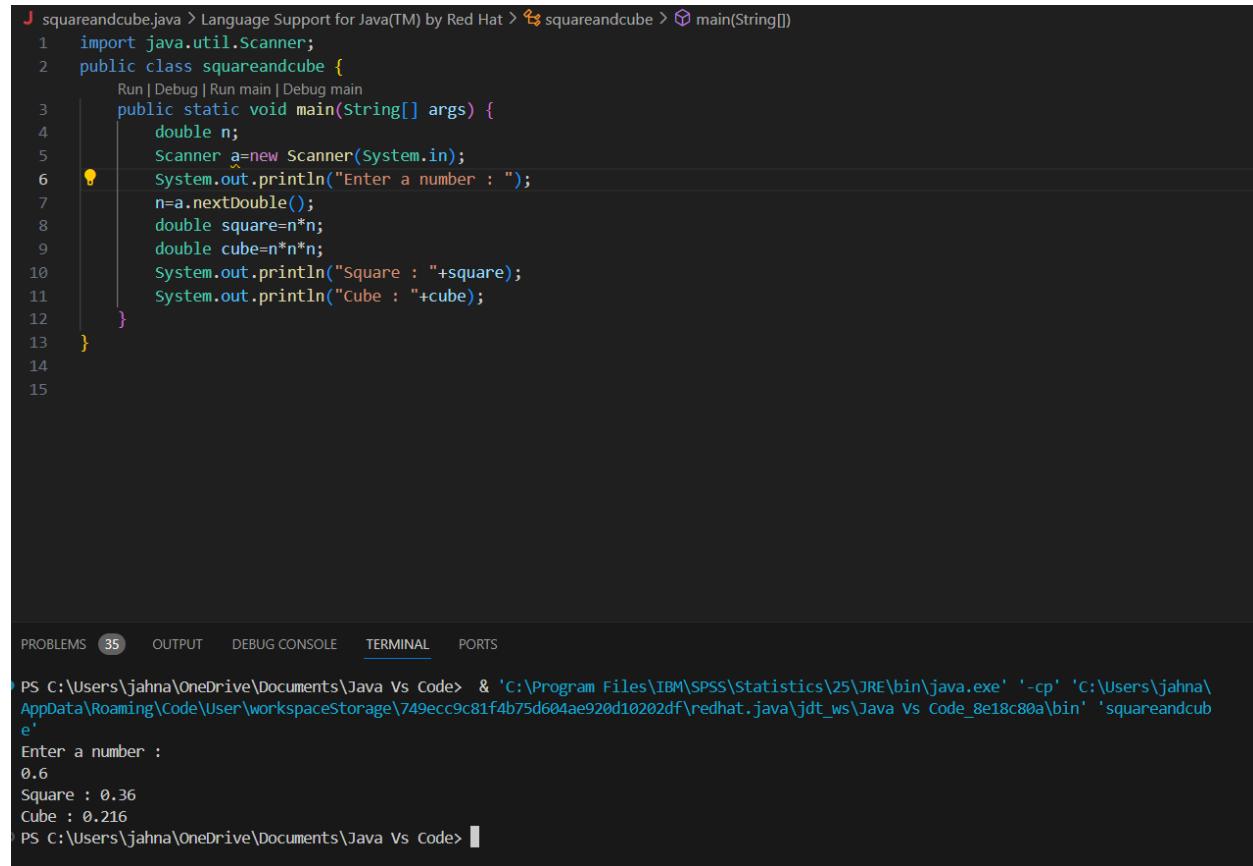
Square Number: 0.36

Cube Number: 0.216

Test cases:

1. 12
2. 0
3. -0.5
4. 14.25
5. -296

OUTPUT:



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

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

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[] = new int[arr.length];
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:

```
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:

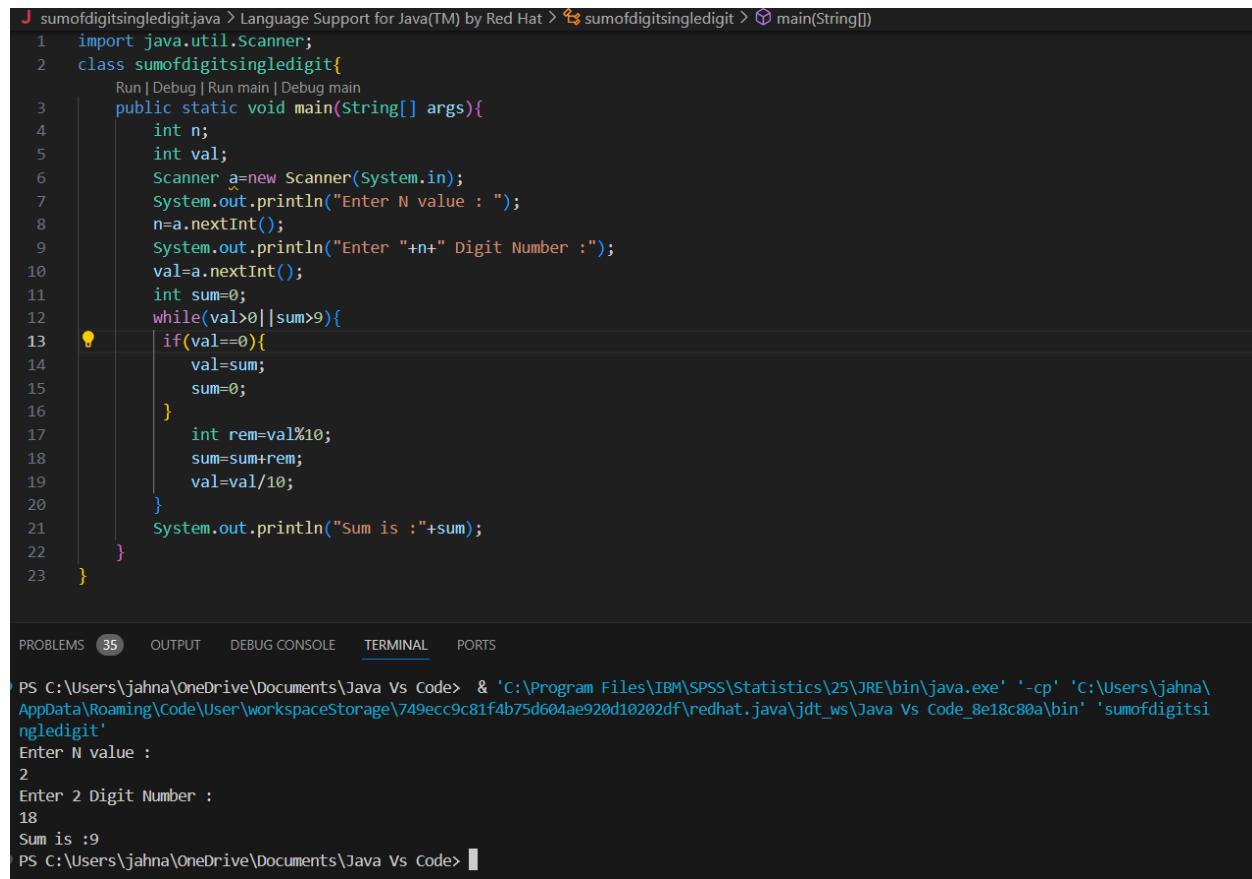
Enter N value: 3

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:

```
J perfectsquare.java > Language Support for Java(TM) by Red Hat > perfectsquare > main(String[])
1 public class perfectsquare {
2     Run | Debug | Run main | Debug main
3     public static void main(String[] args){
4         int n=6561;
5         double sqr=Math.sqrt(n);
6         System.out.println(sqr+(", -"+sqr));
7     }
}
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' '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 is annotated with several yellow lightbulb icons, indicating potential issues or suggestions.

Below the code editor, there is a terminal window showing the following 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\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. A yellow warning icon is positioned next to line 11. The code is as follows:

```
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
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS (35), OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), and PORTS.

The TERMINAL tab shows the following 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\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:

```
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 }
```

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[] args)
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>