

WEEK-7

Problems Based on if statement/Looping/Array in JAVA

1# Write a Java function to implement binary search.

The screenshot shows an IDE interface with several tabs at the top: ...va, PrimeNo_method.java, MaxInArray_method.java, BubSort_method.java, Prac5_10.java, and Prac5_10ViceVersa.java. The main window displays Java code for a binary search algorithm. The code is as follows:

```

7 import java.util.*;
8 public class Week7_1 {
9
10    public static void main(String[] args) {
11        Scanner sc = new Scanner(System.in);
12        System.out.println("Enter number of Elements in array:");
13        int n = sc.nextInt();
14        int a[] = new int[n];
15        System.out.println("Enter the elements (in sorted order):");
16        for(int i = 0; i < n; i++) {
17            a[i] = sc.nextInt();
18        }
19
20        System.out.println("Enter the number you want to find: ");
21        int x = sc.nextInt();
22
23        int s = 0;
24        int e = n - 1;
25        int m;
26
27        while(s <= e) {
28            m = (s + e) / 2; // correct mid calculation
29
30            if(x == a[m]) {
31                System.out.println("Element found at index: " + m);
32                return; // exit the program when found (no flag needed)
33            }
34            else if(x < a[m])
35                e = m - 1;
36            else
37                s = m + 1;
38        }
39        System.out.println("Element not found");
40    }
41 }

```

The code is annotated with a yellow highlighter, specifically highlighting the line `e = m - 1;`. Below the code, the run output window shows the following interaction:

```

week7_1.Week7_1 > main > while (s <= e) > if (x == a[m]) else if (x < a[m]) >
Output - Week7_1 (run) #2 >
run:
Enter number of Elements in array:
5
Enter the elements (in sorted order):
10 20 30 40 50
Enter the number you want to find:
50
Element found at index: 4
BUILD SUCCESSFUL (total time: 30 seconds)

```

The screenshot shows an IDE interface with the following details:

- Top Bar:** Shows tabs for various Java files: ...va, Week6_9.java, Week7_2.java (highlighted in blue), Week7_3.java, Week7_4.java, Week7_5.java, Week7_6.java, and Week7_7.java.
- Source Editor:** Displays Java code for a class named Week7_2. The code implements a bubble sort algorithm to sort an array of integers entered by the user. The line `System.out.println("Enter the elements:");` is highlighted in yellow.
- Output Window:** Titled "Output - Week7_1 (run) #2", it shows the execution of the program. It prompts the user for the number of elements (5) and the elements themselves (30 20 50 40 10). It then displays the sorted array (10 20 30 40 50) and concludes with "BUILD SUCCESSFUL (total time: 14 seconds)".
- Taskbar:** Shows the Windows taskbar with icons for File Explorer, Edge browser, Task View, and others.

3# Write a program to reverse a given string.

2# Write a Java function to arrange the elements of an array in ascending order (Sorting).

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

- Code Editor:** Displays the Java code for Week7_3.java. The code reads a string from the user and prints it reversed.
- Output Window:** Shows the execution results:
 - run:
 - Enter a String
 - polythene
 - enehtylop
 - BUILD SUCCESSFUL (total time: 28 seconds)

4# Write a program to check whether a given string is palindrome or not.

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

- Code Editor:** Displays the Java code for "Week7_4". The code reads a string from the user and checks if it is a palindrome.
- Output Window:** Shows the execution results:
 - run:
 - Enter the string
 - afifa
 - palindrome
 - BUILD SUCCESSFUL (total time: 4 seconds)

5# Write a program to implement factorial of a number through recursion.

The screenshot shows an IDE interface with multiple tabs at the top: Week6_9.java, Week7_2.java, Week7_3.java, Week7_4.java, Week7_5.java (which is the active tab), and Week7_6.java. The main window displays the Java code for Week7_5. The code defines a class Week7_5 with a static method fact that calculates the factorial of a number. It also contains a main method that prompts the user for input and prints the result. The code is highlighted in various colors (blue for keywords, green for comments, red for strings, etc.). The output window at the bottom shows the execution of the program, where the user enters '5' and the program outputs '120'. A status bar at the bottom right indicates a build time of 4 seconds.

```
5  import java.util.*;
6  public class Week7_5 {
7
8      public static int fact(int n){
9          if(n==0){
10              return 1;
11          }
12          return n*fact(n-1);
13      }
14
15      public static void main(String[] args){
16          Scanner sc = new Scanner(System.in);
17          System.out.println("Enter number of Elements in array:");
18          int n = sc.nextInt();
19
20          System.out.println(fact(n));
21      }
22  }
```

Week7_5 > main >

Output - Week7_1 (run) #2 >

```
run:
Enter number of Elements in array:
5
120
BUILD SUCCESSFUL (total time: 4 seconds)
```

6# Write a program to implement Fibonacci series of a number with and without recursion.

Without Recursion

The screenshot shows an IDE interface with multiple tabs at the top: ...va, Prac6_2.java, Prac6_4.java, Prac6_5.java, Prac6_1.java, and Week6. The main window displays a Java source code file named sessionall_1.java. The code defines a class Week6_9 with a static method fibonacci that prints the first N numbers of the Fibonacci sequence. The output window below shows the results of four runs, with the last run (#4) showing the values 5, 8, 13, 21, and 34.

```
5
package sessionall_1;
6 import java.util.Scanner;
7 public class Week6_9 {
8
9     public static void fibonacci(int N){
10         int first=0;
11         int second=1;
12         int next;
13
14         for(int i=1; i<=N; i++){
15             System.out.println(first+" ");
16             next=first+second;
17             first=second;
18             second=next;
19         }
20     }
21 }
```

sessionall1_Week6_9 > main >

Output x

< run x Week66 (run) #2 x Week66 (run) #3 x Week66 (run) #4 x W

5
8
13
21
34

BUILD SUCCESSFUL (total time: 6 seconds)

With Recursion

The screenshot shows a Java code editor with a file named Week7_6.java. The code implements a recursive function to calculate the Fibonacci series. It includes a main method that reads an integer input from the user and prints the series up to that number. The output window shows the execution of the program, including the user input and the resulting Fibonacci sequence.

```
5 import java.util.*;
6 public class Week7_6 {//with recursion
7     public static int fibonacci(int n){
8         if(n==0)
9             return 0;
10        else if(n==1)
11            return 1;
12        return(fibonacci(n-1)+fibonacci(n-2));
13    }
14
15    public static void main(String[] args){
16        Scanner sc=new Scanner(System.in);
17        System.out.println("Enter a number n: ");
18        int n=sc.nextInt();
19        System.out.println("The Fibonacci series is: ");
20        for(int i=0; i<n; i++)
21            System.out.print(fibonacci(i)+" ");
22    }
}
```

Output:

- Week7_1 (run) #2
- Week7_1 (run) #3

```
run:
Enter a number n:
6
The Fibonacci series is:
0 1 1 2 3 5 BUILD SUCCESSFUL (total time: 2 seconds)
```

Optional:

7# Write a Java function to find the greatest common divisor (GCD) of two numbers with and without using recursion.

Without recursion

```

5 import java.util.*;
6 public class Week7_7WOR {
7     public static void main(String[] args) {
8         Scanner sc=new Scanner(System.in);
9         System.out.println("Enter the first number: ");
10        int a=sc.nextInt();
11        System.out.println("Enter the second number: ");
12        int b=sc.nextInt();
13
14        int GCD=0;
15        for(int i=12; i>=1; i--) {
16            if((b%i==0) && (a%i==0)) {
17                GCD=i;
18                break;
19            }
20        }
21        System.out.println(GCD);

```

The code uses a for loop to iterate from 12 down to 1. It checks if both numbers are divisible by the current iteration value. If they are, it sets that value as the GCD and breaks out of the loop. Finally, it prints the GCD.

With recursion

```

5 import java.util.*;
6 public class Week7_7WR {
7
8     public static int GCD(int n, int d) {
9         if(n%d==0)
10             return d;
11         return(GCD(n, n%d));
12     }
13     public static void main(String [] args) {
14         Scanner sc=new Scanner(System.in);
15         System.out.println("Enter the first number:");
16         int n=sc.nextInt();
17         System.out.println("Enter the second number:");
18         int d=sc.nextInt();
19         System.out.println(GCD(n, d));
20     }

```

The code defines a recursive method GCD that takes two integers n and d as parameters. If n is divisible by d (n%d==0), it returns d. Otherwise, it calls itself with n and n%d as arguments. Finally, it prints the result of the GCD method in the main function.

8# Write a program to check whether two strings are anagrams of each other ("listen" and "silent" are anagrams).

The screenshot shows a Java IDE interface with multiple tabs at the top: Week7_7WOR.java, Week6_11.java, Week7_8.java (which is the active tab), Notebook.java, and Fact_method.java. Below the tabs is a toolbar with various icons. The main area displays the source code for Week7_8.java:

```
5 import java.util.*;
6 public class Week7_8 {
7
8     public static void main(String[] args){
9         Scanner sc=new Scanner(System.in);
10        System.out.println("Enter the first String:");
11        String s=sc.nextLine();
12        System.out.println("Enter the second String:");
13        String r=sc.nextLine();
14        int a[]=new int[256];
15
16        if(s.length() != r.length()){
17            System.out.println("Not anagram");
18            return;
19        }
20        for(int i=0; i<s.length(); i++){
21            int e=s.charAt(i);
```

9# Write a Java program to implement Fibonacci series up to N terms
(0,1,1,2,3,5....).

The screenshot shows a Java IDE interface with the following details:

- Title Bar:** Shows multiple open files: ...va, MaxinArray_method.java, BubSort_method.java, Prac5_10.java, Prac5_10ViceVersajava, and Week7_9.
- Source Editor:** Displays the `Week7_9` class with its implementation of the Quicksort algorithm. The code includes methods `quicksort`, `part`, and `main`.
- Output Window:** Shows the execution of the program. It prompts for the number of elements (5), enters five numbers (10 30 40 20 50), displays the sorted array (10 20 30 40 50), and concludes with "BUILD SUCCESSFUL (total time: 25 seconds)".

```
5 import java.util.*;
6 public class Week7_9 {
7
8     public static void quick(int[] arr, int s, int e) {
9         if (s < e) {
10             int p = part(arr, s, e);
11             quick(arr, s, p - 1);
12             quick(arr, p + 1, e);
13         }
14     }
15
16     public static int part(int [] arr, int s, int e){
17         int key = arr[e];
18         int i = s - 1;
19
20         for (int j = s; j < e; j++) {
21             if (arr[j] < key) {
22                 i++;
23                 int temp = arr[i];
24                 arr[i] = arr[j];
25                 arr[j] = temp;
26             }
27         }
28
29         int temp = arr[i + 1];
30         arr[i + 1] = arr[e];
31         arr[e] = temp;
32
33         return i + 1;
34     }
35
36     public static void main(String[] args) {
37         Scanner input = new Scanner(System.in);
38
39         System.out.print("Enter number of elements: ");
40         int n = input.nextInt();
41         int[] arr = new int[n];
42
43         System.out.println("Enter " + n + " numbers:");
44         for (int i = 0; i < n; i++) {
45             arr[i] = input.nextInt();
46         }
47
48         quick(arr, 0, n - 1);
49
50         System.out.println("Sorted array:");
51         for (int num : arr) {
52             System.out.print(num + " ");
53         }
54
55         input.close();
56     }
57 }
```

Week7_9

Output X

Week7_1 (run) #2 X Week7_1 (run) #3 X

run:
Enter number of elements: 5
Enter 5 numbers:
10 30 40 20 50
Sorted array:
10 20 30 40 50 BUILD SUCCESSFUL (total time: 25 seconds)