Daily Program Practice   Date- 23/11/2023

Program 1: Maximum Width of Binary Tree

```java
package ishwarchavan.com;

class Node {   //left and right node created
	int data;
	Node left, right;

	Node(int item)
	{
		data = item;
		left = right = null;
	}
}

public class MaxWidthOfBinaryTree  {    //function created
	Node root;

	int getMaxWidth(Node node)        /* Function to get the maximum width of a binary 
tree*/
	{
		int maxWidth = 0;
		int width;
		int h = height(node);
		int i;

		for (i = 1; i <= h; i++) {       /* Get width of each level and compare 
			the width with maximum width so far */
			width = getWidth(node, i);
			if (width > maxWidth)
				maxWidth = width;
		}

		return maxWidth;
	}


	int getWidth(Node node, int level)    /* Get width of a given level */
	{
		if (node == null)
			return 0;

		if (level == 1)
			return 1;
		else if (level > 1)
			return getWidth(node.left, level - 1) + getWidth(node.right, level - 
1);
		return 0;
	}

	int height(Node node)
	{
		if (node == null)
			return 0;
		else {

			int lHeight = height(node.left);   /* compute the height of each 
subtree */
			int rHeight = height(node.right);

			return (lHeight > rHeight) ? (lHeight + 1) : (rHeight + 1);    /* use 
the larger one */
		}
	}
	public static void main(String args[])   //main program started
	{
		MaxWidthOfBinaryTree  tree = new MaxWidthOfBinaryTree ();

		tree.root = new Node(1);
		tree.root.left = new Node(2);
		tree.root.right = new Node(3);
		tree.root.left.left = new Node(4);
		tree.root.left.right = new Node(5);
		tree.root.right.right = new Node(8);
		tree.root.right.right.left = new Node(6);
		tree.root.right.right.right = new Node(7);

		System.out.println("Maximum width is " + tree.getMaxWidth(tree.root));  //
Function call
	}
}
```

Program 2: Merge Two 2D arryas

```
package ishwarchavan.com;

import java.util.Arrays;

public class MergeTwoArrays {
	public static void main(String[] args) {

		int[] a = { 10, 20, 30, 40 };   // first array

		int[] b = { 50, 60, 70, 80 };   // second array

		int a1 = a.length;      // determines length of firstArray

		int b1 = b.length;      // determines length of secondArray

		int c1 = a1 + b1;

		int[] c = new int[c1];   // create the resultant array

		System.arraycopy(a, 0, c, 0, a1);   // using the pre-defined function
arraycopy
		System.arraycopy(b, 0, c, a1, b1);

		System.out.println(Arrays.toString(c));    // prints the resultant array
	}
}
```

Program 3: Add Binary

```
package ishwarchavan.com;

public class AddBinary {  //class created

	static String add_Binary(String x, String y){  // Function to add two binary
strings

		int num1 = Integer.parseInt(x, 2);   // converting binary string into
integer(decimal number)

		int num2 = Integer.parseInt(y, 2);
		int sum = num1 + num2;  // Adding those two decimal numbers and storing in
sum

		String result = Integer.toBinaryString(sum);
		return result;
	}
	public static void main(String args[]) {   //main program started
```

```
		String x = "011011", y = "1010111";

		System.out.print(add_Binary(x, y));
	}
}
```

Program 4: Happy Number

```
package ishwarchavan.com;

public class HappyNumber{  //class created

static int numSquareSum(int n) {   //function created

	int squareSum = 0;
	while (n!= 0)   //checking condition
	{
		squareSum += (n % 10) * (n % 10);
		n /= 10;
	}
	return squareSum;
}

static boolean isHappynumber(int n)  //method return true if n is Happy number
{
	int slow, fast;

	slow = fast = n;  // initialize slow and fast by n
	do
	{
		slow = numSquareSum(slow);  // move slow number by one iteration

		fast = numSquareSum(numSquareSum(fast));  // move fast number by two
iteration
	}
	while (slow != fast);  // if both number meet at 1,then return true
	return (slow == 1);
}

public static void main(String[] args) {   //main program started

	int n = 13;
	if (isHappynumber(n))
		System.out.println(n +
		" is a Happy number");
	else
		System.out.println(n +
		" is not a Happy number");
}
}
```

Program 5: Fizz Buzz

```
package ishwarchavan.com;

import java.util.ArrayList;
import java.util.List;

public class FizzBuzz {   //class created
	public static List<String> fizzBuzz(int n){  //function created

		List<String> result = new ArrayList<>();  // Declare a list of strings to
store the results

		for (int i = 1; i <= n; ++i) {  // Loop from 1 to n (inclusive)
```

```
                if (i % 3 == 0 && i % 5 == 0) {  // Check if i is divisible by both 3
and 5

                    result.add("FizzBuzz");  // Add "FizzBuzz" to the result list
                }

                else if (i % 3 == 0) {                    // Check if i is divisible
by 3

                    result.add("Fizz");    // Add "Fizz" to the result list
                }

                else if (i % 5 == 0) {    // Check if i is divisible by 5

                    result.add("Buzz");  // Add "Buzz" to the result list
                }
                else {

                    result.add(Integer.toString(i));  // Add the current number as
a string to the result list
                }
            }
            return result;  // Return the result list
        }

        public static void main(String[] args){
            int n = 20;

            List<String> result = fizzBuzz(n);  // Call the fizzBuzz function to get
the result

            for (String s : result) {      // Print the result
                System.out.print(s + " ");
            }
        }
}
```

Program 6: Check if Number is sum of power of three

```
package ishwarchavan.com;

public class CheckPowerOfThree {  //class created
public static void DistinctPowersOf3(int N){   //function created

        while (N > 0) {   // Iterate until N is non-zero

            if (N % 3 == 2) {  // Termination Condition
                System.out.println("No");
                return;
            }
            N /= 3;    // Right shift ternary bits by 1 for the next digit
        }

        System.out.println("Yes");    // If N can be expressed as the sum of perfect
powers of 3
}

public static void main(String args[]) {  //main program created
        int N = 12;
        DistinctPowersOf3(N);
}
}
```

Program 7: Check if Number is sum of power of three

```java
package ishwarchavan.com;
public class IntegerToWord {  //class created

	static String numberToWords(long n){  //function created
		long limit = 1000000000000L, curr_hun, t = 0;

		if (n == 0)   // If zero return zero
			return ("Zero");

		String multiplier[] = { "", "Trillion", "Billion","Million", "Thousand" };
// Array to store the powers of 10

		String first_twenty[] = {"",	"One",	"Two",	"Three","Four",
"Five",	"Six",	"Seven","Eight", "Nine",	"Ten",	"Eleven","Twelve",
"Thirteen", "Fourteen", "Fifteen","Sixteen", "Seventeen", "Eighteen", "Nineteen"};

		String tens[] = { "",	"Twenty", "Thirty","Forty", "Fifty",
"Sixty","Seventy", "Eighty", "Ninety" };  // Array to store multiples of ten

		if (n < 20L)   //checking condition
			return (first_twenty[(int)n]);
		String answer = "";
		for (long i = n; i > 0; i %= limit, limit /= 1000) {   //loop iterating
			curr_hun = i / limit;

			while (curr_hun == 0) {   //condition checkig
				i %= limit;

				limit /= 1000;    // Divide the limit by 1000, shifts the
multiplier

				curr_hun = i / limit;

				++t;  // Shift the multiplier
			}

			if (curr_hun > 99)   //if true then execute below statement
				answer += (first_twenty[(int)curr_hun / 100]+ " Hundred ");

			curr_hun = curr_hun % 100;  // Bring the current hundred to tens

			if (curr_hun > 0 && curr_hun < 20)        // If the value in tens
belongs to [1,19], add using the first_twenty
				answer += (first_twenty[(int)curr_hun] + " ");

			else if (curr_hun % 10 == 0 && curr_hun != 0)
				answer += (tens[(int)curr_hun / 10 - 1] + " ");

			else if (curr_hun > 20 && curr_hun < 100)
				answer += (tens[(int)curr_hun / 10 - 1] + " "+
first_twenty[(int)curr_hun % 10] + " ");

			if (t < 4)    // If Multiplier has not become less than 1000,shift it
				answer += (multiplier[(int)++t] + " ");
		}
		return (answer);
	}
	public static void main(String args[]) {   //main program started
		long n = 360;
		System.out.println(numberToWords(n));

		n = 1234;
		System.out.println(numberToWords(n));
	}
}
```

Program 8: Intersection Of Two Linked Lists

```
package ishwarchavan.com;

import java.util.*;
import java.io.*;

public class IntersectioOfTwoLinkedLists {  //class created
      static class Node {
            int data;
            Node next;
            Node(int d)
            {
                  data = d;
                  next = null;
            }
      }

      public Node getIntersectionNode(Node head1, Node head2){  //function created
            while (head2 != null) {   // checking condition
                  Node temp = head1;
                  while (temp != null) {

                        if (temp == head2) {   // if both Nodes are same
                              return head2;
                        }
                        temp = temp.next;
                  }
                  head2 = head2.next;
            }

            return null;
      }

      public static void main(String[] args){   // main program started
            IntersectioOfTwoLinkedLists list = new IntersectioOfTwoLinkedLists();
//object created

            Node head1, head2;

            head1 = new Node(10);
            head2 = new Node(3);

            Node newNode = new Node(6);
            head2.next = newNode;

            newNode = new Node(9);
            head2.next.next = newNode;

            newNode = new Node(15);
            head1.next = newNode;
            head2.next.next.next = newNode;

            newNode = new Node(30);
            head1.next.next = newNode;

            head1.next.next.next = null;

            Node intersectionPoint = list.getIntersectionNode(head1, head2);

            if (intersectionPoint == null) {   //checking condition
                  System.out.print(" No Intersection Point \n");
            }
            else {
                  System.out.print("Intersection Point: "+ intersectionPoint.data);
            }
      }
}
```

Program 9: Factorial Trailing Zeroes

```
package ishwarchavan.com;
import java.io.*;

public class TrailingZeroes {    //class created

      static int findTrailingZeros(int n){   //function created
            if (n < 0) // Negative Number Edge Case
                  return -1;
            int count = 0;   // Initialize result

            for (int i = 5; n / i >= 1; i *= 5)    // Keep dividing n by powers of 5 and 
update count
                  count += n / i;

            return count;
      }
      public static void main(String[] args){   //main program started
            int n = 100;
            System.out.println("Count of trailing 0s in " + n + "! is "+ 
findTrailingZeros(n));
      }
}
```