1.Search element in an array from last index

i/p : arr[] = {20,90,1,90,6}, num = 90

o/p : 3

CODE :

public class searchElementfromLast {

    static int search(int arr[], int index, int num) {

        if (index < 0) {

            return -1;

        }

        if (arr[index] == num) {

            return index;

            //return arr.length-1-index;

        }

        return search(arr, index-1, num);

    }

    public static void main(String[] args) {

        int arr[] = { 20,90,1,90,6 };

        int num = 90;

        int i = arr.length-1;

        int index = search(arr, i, num);

        System.out.println(index);

    }

}

2. Return an array containing all indexes where element occur

i/p : arr[] = {20,5,6,0,5,9,5}, num=5

o/p : [1,4,6]

CODE:

import java.util.ArrayList;

import java.util.List;

public class arrayContainingIndexes {

    public static List<Integer> search(int[] arr, int index, int num) {

        List<Integer> indexes = new ArrayList<>();

        if (index == arr.length) {

            return indexes;

        }

        if (arr[index] == num) {

            indexes.add(index);

        }

        indexes.addAll(search(arr, index + 1, num));

        return indexes;

}

 public static void main(String[] args) {

        // int arr[] = { 20, 5, 6, 0, 5, 9, 5 };

        // int num = 5;

        // List<Integer> indexes = search(arr, 0 , num);

        // System.out.println(indexes);

}

3. Pattern-6

     \*

    \*\*\*

   \*\*\*\*\*

  \*\*\*\*\*\*\*

 \*\*\*\*\*\*\*\*\*

CODE:

public class pattern4 {

    static void printSpace(int noOfSpaces) {

        if(noOfSpaces == 0) {

            return;

        }

        System.out.print(" ");

        printSpace(noOfSpaces-1);

    }

    static void printStar(int noOfStar) {

        // Base Case

        if(noOfStar == 0) {

            return;

        }

        // Processing Logic

        System.out.print("\*");

        // Small Problem

        printStar(noOfStar - 1);

    }

    static void printPattern(int rows, int currentRow) {

        if(rows == 0) {

            return;

        }

        printSpace(rows-1);

        printStar(currentRow\*2 -1);  // print the row

        System.out.println();   // move to the new line

        printPattern(rows-1, currentRow+1);

    }

    public static void main(String[] args) {

        printPattern(5, 1);

    }

}

4. Pattern-4

1

2 3

4 5 6

7 8 9 10

CODE :

public class pattern5 {

    static int currentNumber = 1;

    static void printPattern(int rows) {

        if (rows == 0) {

            return;

        }

        printPattern(rows - 1);

        printNumbersForRow(rows);

        System.out.println();

    }

    static void printNumbersForRow(int row) {

        if (row == 0) {

            return;

        }

        printNumbersForRow(row - 1);

        System.out.print(currentNumber + " ");

        currentNumber++;

    }

    public static void main(String[] args) {

        int numRows = 4;

        printPattern(numRows);

    }

}

5.Pattern-5

A

BB

CCC

DDDD

EEEEE

CODE:

public class pattern6 {

    void printTriangle(int n) {

        // code her

        for (int i = 0; i < n; i++) {

            char ch = (char) ('A' + i);

            for (int j = 0; j <= i; j++) {

                System.out.print(ch);

            }

            System.out.println();

        }

    }

    public static void main(String[] args) {

        int numRows = 5;

        pattern6 p = new pattern6();

        p.printTriangle(numRows);

    }

}