C-DAC Mumbai

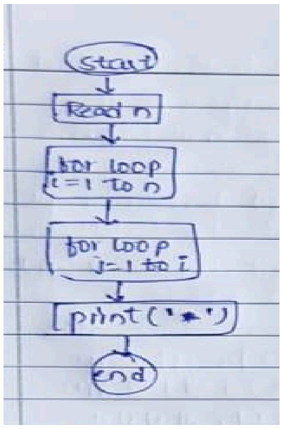
Date 26/09/2024

Subject: Algorithm and Data Structure
Assignment 2

1. Printing Patterns
Problem: Write a Java program to print patterns such as a right triangle of stars.

```
class Pattern{
	public static void pattern(int n){
		for(int i=1;i<=n;i++){
			for(int j=1;j<=i;j++){
				System.out.print("* ");
			}
			System.out.println();
		}
	}
	public static void main(String args[]){
		pattern(3);
		pattern(5);
	}
}
```

```
C:\Users\saira\OneDrive\Documents\CDAC\ADS Module\Assignment2>java Pattern
*
* *
* * *
*
* *
* * *
* * * *
* * * * *
```



Time Complexity: $O(n^2)$

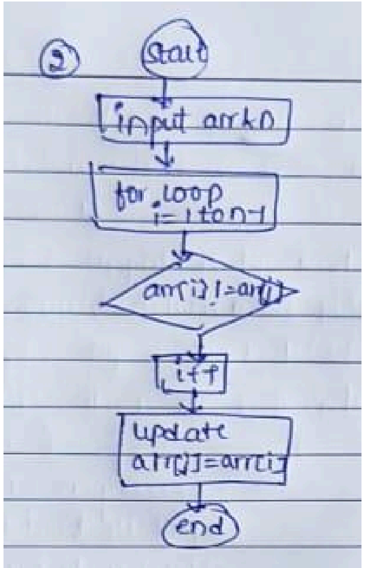
Space Complexity: O(1)

2. Remove Array Duplicates

Problem: Write a Java program to remove duplicates from a sorted array and return the new length of the array.

```
class Duplicates {
    static int remove(int arr[], int n) {
        if (n == 0 || n == 1) {
            return n;
        }

        int j = 0;

        for (int i = 1; i < n; i++) {
            if (arr[i] != arr[j]) {
                j++;
                arr[j] = arr[i];
            }
        }
        return j + 1;
    }

    public static void main(String args[]) {
        int[] arr1 = {1, 1, 2};
        int[] arr2 = {0, 0, 1, 1, 2, 2, 3, 3};
        int l1 = remove(arr1, arr1.length);
        System.out.println("New length for arr1: " + l1);
        int l2 = remove(arr2, arr2.length);
        System.out.println("New length for arr2: " + l2);
    }
}
```

```
C:\Users\saira\OneDrive\Documents\CDAC\ADS Module\Assignment2>java Duplicates
New length for arr1: 2
New length for arr2: 4
```

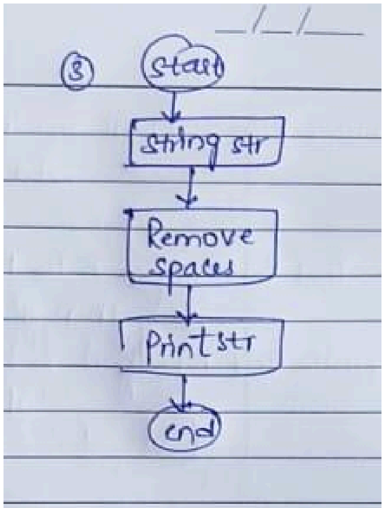


Time Complexity: O(n)

Space Complexity: O(1)

3. Remove White Spaces from String
Problem: Write a Java program to remove all white spaces from a given string.

```
class Spaces{
        public static void main(String args[]){
                String str1="Hello World";
                String str2="Java Programming";
                str1=str1.replaceAll("\\s","");
                str2=str2.replaceAll("\\s","");
                System.out.println(str1);
                System.out.println(str2);
        }
}
```



```
C:\Users\saira\OneDrive\Documents\CDAC\ADS Module\Assignment2>java Spaces
HelloWorld
JavaProgramming
```

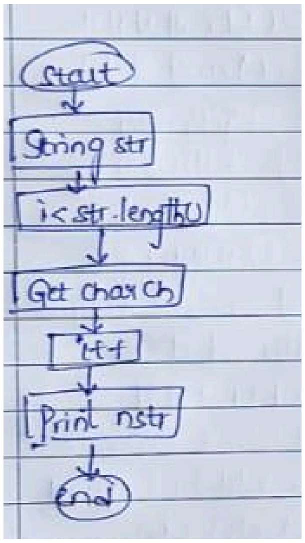
Time Complexity: O(n)

Space Complexity: O(1)

4. Reverse a String
Problem: Write a Java program to reverse a given string.

```
class Reverse{
        public static void reverse(String str){
                char ch;
                String nstr="";
                for(int i=0;i<str.length();i++){
                        ch=str.charAt(i);
                        nstr=ch+nstr;
                }
                System.out.println(nstr);
        }
        public static void main(String args[]){
                reverse("hello");
                reverse("Java");
        }
}
```

```
C:\Users\saira\OneDrive\Documents\CDAC\ADS Module\Assignment2>java Reverse
olleh
avaJ
```



Time Complexity: O(n)

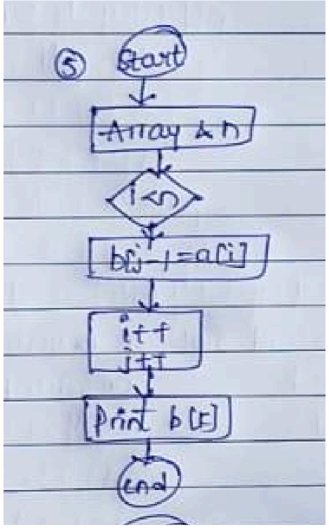
Space Complexity: O(n)

5. Reverse Array in Place

Problem: Write a Java program to reverse an array in place.

```
class ArrayReverse{
        static void reverse(int a[],int n){
                int[] b = new int[n];
                int j=n;
                for(int i=0;i<n;i++){
                        b[j-1]=a[i];
                        j=j-1;
                }
                for(int k=0;k<n;k++){
                        System.out.print( b[k] +" " );
                }
                System.out.println();
        }
        public static void main(String args[]){
                int[] arr = {1, 2, 3, 4};
                int[] arr1={7,8,9};
                reverse(arr,arr.length);
                reverse(arr1,arr1.length);
        }
}
```

```
C:\Users\saira\OneDrive\Documents\CDAC\ADS Module\Assignment2>java ArrayReverse
4 3 2 1
9 8 7
```



Time Complexity: O(n)

Space Complexity: O(n)

6. Reverse Words in a String

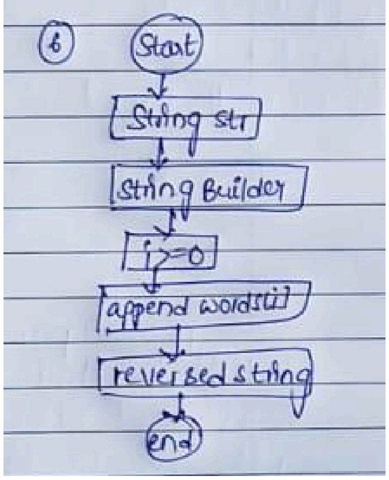
Problem: Write a Java program to reverse the words in a given sentence.

```
public class Words{
        static void reverse(String str){
                String[] words = str.split(" ");
                StringBuilder reversestring = new StringBuilder();

                for(int i=words.length-1;i>=0;i--){
                        reversestring.append(words[i]).append(" ");
                }
                System.out.println(reversestring.toString().trim());
        }
        public static void main(String args[]){
                reverse("Hello World");
                reverse("Java Programming");
        }
}
```

```
C:\Users\saira\OneDrive\Documents\CDAC\ADS Module\Assignment2>java Words
World Hello
Programming Java
```

Time Complexity: O(n)

Space Complexity: O(n)



7. Reverse a Number

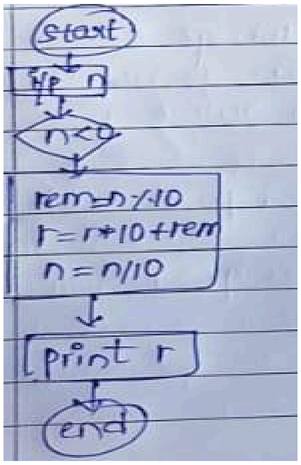
Problem: Write a Java program to reverse a given number.

```
class Number{
	public static void reverse(int n){
		int r=0;
		while(n!=0){
			int rem = n%10;
			r=r*10+rem;
			n=n/10;
		}
		System.out.println(r);
	}
	public static void main(String args[]){
		reverse(12345);
		reverse(-9876);
	}
}
```

```
C:\Users\saira\OneDrive\Documents\CDAC\ADS Module\Assignment2>java Number
54321
-6789
```

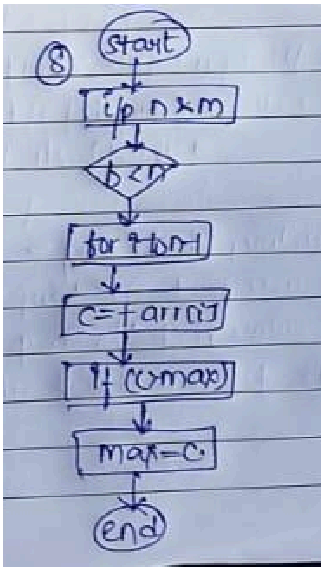
Time Complexity: O(n)

Space Complexity: O(1)



8. Array Manipulation
Problem: Perform a series of operations to manipulate an array based on range update queries. Each query adds a value to a range of indices.

```
import java.util.Scanner;
public class ArrayManipulation {
    public static long arrayManipulation(int n, int[][] queries) {
        long[] arr = new long[n + 1];

        for (int i = 0; i < queries.length; i++) {
            int a = queries[i][0];
            int b = queries[i][1];
            int k = queries[i][2];

            arr[a - 1] += k;
            if (b < n) {
                arr[b] -= k;
            }
        }

        long max = 0;
        long current = 0;

        for (int i = 0; i < n; i++) {
            current += arr[i];
            if (current > max) {
                max = current;
            }
        }

        return max;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        System.out.print("Enter the number of queries: ");
        int m = scanner.nextInt();

        int[][] queries = new int[m][3];

        for (int i = 0; i < m; i++) {
            queries[i][0] = scanner.nextInt();
            queries[i][1] = scanner.nextInt();
            queries[i][2] = scanner.nextInt();
        }

        long result = arrayManipulation(n, queries);
        System.out.println(result);

    }
}
C:\Users\saira\OneDrive\Documents\CDAC\ADS Module\Assignment2>java ArrayManipulation
5
Enter the number of queries: 3
1 2 100
2 5 100
3 4 100
200
C:\Users\saira\OneDrive\Documents\CDAC\ADS Module\Assignment2>java ArrayManipulation
4
Enter the number of queries: 2
1 3 50
2 4 70
120
```



Time Complexity: O(n+m)

Space Complexity: O(n)

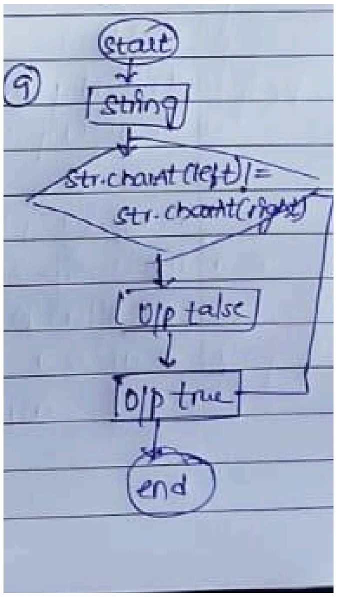
9. String Palindrome

Problem: Write a Java program to check if a given string is a palindrome.

```
public class Palindrome {
    public static boolean isPalindrome(String str) {
        int left = 0;
        int right = str.length() - 1;
        while (left < right) {
            if (str.charAt(left) != str.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }

    public static void main(String[] args) {
        String str1 = "madam";
        System.out.println(isPalindrome(str1));

        String str2 = "hello";
        System.out.println(isPalindrome(str2));
    }
}
C:\Users\saira\OneDrive\Documents\CDAC\ADS Module\Assignment2>java Palindrome
true
false
```

Time Complexity: O(n)

Space Complexity: O(1)



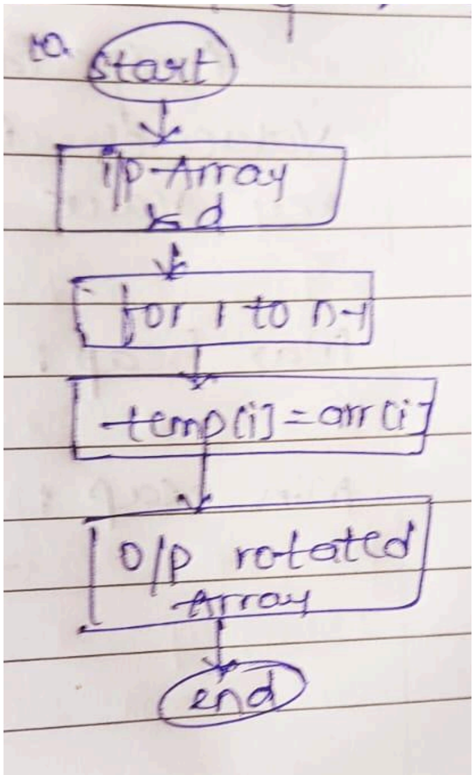
10. Array Left Rotation
Problem: Write a Java program to rotate an array to the left by d positions.

```
import java.util.Arrays;

public class ArrayLeftRotation {
    public static void rotateLeft(int[] arr, int d) {
        int n = arr.length;
        int[] temp = new int[n];

        for (int i = 0; i < n; i++) {
            temp[i] = arr[(i + d) % n];
        }
        for (int i = 0; i < n; i++) {
            arr[i] = temp[i];
        }
    }

    public static void main(String[] args) {
        int[] arr1 = {1, 2, 3, 4, 5};
        int d1 = 2;
        rotateLeft(arr1, d1);
        System.out.println(Arrays.toString(arr1));

        int[] arr2 = {10, 20, 30, 40};
        int d2 = 1;
        rotateLeft(arr2, d2);
        System.out.println( Arrays.toString(arr2));
    }
}
```

```
C:\Users\saira\OneDrive\Documents\CDAC\ADS Module\Assignment2>java ArrayLeftRotation
[3, 4, 5, 1, 2]
[20, 30, 40, 10]
```

Time Complexity: O(n)

Space Complexity: O(1)


10.
Start
i/p Array
&d
for i to n-1
temp[i] = arr[i]
O/P rotated
Array
end