



IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING
(An Autonomous Institute affiliated to JNTUH, Hyderabad)
Dundigal, Hyderabad - 500 043

LABORATORY WORK BOOK

Name of the Student : N. Ravi Chandarka

Class : CSD-B Semester : IIIrd Semester

Course Code : ACSD11 Course Name : DS

Name of the Course Faculty : Dr. G. Sucharitha Reddy Faculty ID : IAREID805

Exercise Number : 1 Week Number : 1 Date : 03/09/2024

Roll Number									
5	3	9	5	1	A	6	7	B	3

Exercise Number			Week Number							
S. No.	Exercise Number	EXERCISE NAME	MARKS AWARDED							Total
			Aim/ Preparation	Algorithm / Procedure		Source Code	Program Execution	Viva - Voce		
				Performance in the Lab		Calculations and Graphs	Results and Error Analysis			
			4	4		4	4	4	20	
1	1.1	Sum of last digits of 2 given numbers	4	2	2	4	3	4	19	
2	1.2	Is N an exact multiple of M2	4	2	2	4	3	4	19	
3	1.3	Combine Strings	4	2	2	4	3	3	18	
4	1.4	Even or odd	4	2	2	4	3	3	18	
5	1.5	Second last digit of a given number	4	2	2	4	4	3	19	
6	1.6	Alternate string Combiner	4	2	2	4	4	3	19	
7	1.7	Padovan Sequence	4	2	2	4	4	3	19	
8	1.8	Leaders in an Array	4	2	2	4	4	4	20	
9	1.9	Find value of a number raised to its reverse	4	2	2	4	4	4	20	
10	1.10	Mean of Array using Recursion	4	2	2	4	4	3	19	
11										
12										

Signature of the Student

Signature of the Faculty

START WRITING FROM HERE

- 1.1 Aim: Rohit wants to add the last digits of two given numbers.
For example, If the given numbers are 267 and 154, the output should be 11. Write a program to help Rohit achieve this for any given two numbers.

Code:

```
import java.util.Scanner;

class AddLastDigitsFunction {
    int addLastDigits(int n1, int n2) {
        int lastDigit1 = Math.abs(n1) % 10;
        int lastDigit2 = Math.abs(n2) % 10;
        return lastDigit1 + lastDigit2;
    }

    public static void main(String args[]) {
        AddLastDigitsFunction obj = new AddLastDigits();
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter First number: ");
        int n1 = scanner.nextInt();
        System.out.print("Enter Second number: ");
        int n2 = scanner.nextInt();
        System.out.println(obj.addLastDigits(n1, n2));
        scanner.close();
    }
}
```

Input: 10 5, 154

(3) 207, 154

① 207, 154

② 207, 154

Output: ① 11

② 11

③ 11

④ 11

Aim: Write a function that accepts two parameters and find whether the first parameter is an exact multiple of the second parameter. If it is exact the function should return required number, else return 1.

Code:

```

public class MultipleChecker {
    public static int checkMultiple(int num1, int num2) {
        if (num1 == 0 || num2 == 0) {
            return 0;
        }
        if (num1 % num2 == 0) {
            return num1;
        }
        else {
            return 1;
        }
    }

    public void main (String[] args) {
        int num1 = 10;
        int num2 = 5;
    }
}

```



```

int result = checkMultiple(num1, num2);
System.out.println("Input: num1 = " + num1 + ", num2 = " + num2);
System.out.println("Output: " + result);

num1 = -10;
result = checkMultiple(num1, num2);
System.out.println("Input: num1 = " + num1 + ", num2 = " + num2);
System.out.println("Output: " + result);

num1 = 0;
result = checkMultiple(num1, num2);
System.out.println("Input: num1 = " + num1 + ", num2 = " + num2);
System.out.println("Output: " + result);

num1 = 10;
num2 = 3;
result = checkMultiple(num1, num2);
System.out.println("Input: num1 = " + num1 + ", num2 = " + num2);
System.out.println("Output: " + result);

```

3

3

Input: ① num1 = 10, num2 = 5

② num1 = 10, num2 = 3

Output: ① 2

② 1.

- 1.3 Aim: Given 2 strings a and b, return a new string of the form Short + Long + Short, with the shorter string on the outside and the longer string in the inside.

Code:

```
import java.util.Scanner;
```

```
public class StringCombiner {
```

```
    public static String combineStrings(String a, String b) {
```

```
        String shortStr = a.length() < b.length() ? a : b;
```

```
        String longStr = a.length() > b.length() ? a : b;
```

```
        return shortStr + longStr + shortStr;
```

```
    }
```

```
    public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter the first String: ");
```

```
        String a = scanner.nextLine();
```

```
        System.out.print("Enter the second String: ");
```

```
        String b = scanner.nextLine();
```

```
        String result = combineStrings(a, b);
```

```
        System.out.println("Combined Strings: " + result);
```

```
        scanner.close();
```

```
    }
```

```
}
```

Enter first String: "hi"

Input: Enter second String: "hello"

Output: "hihellohi"

141. Write a function that accepts 6 input parameters. The first 5 input parameters are of type int, the sixth input parameter is of type String. If the sixth parameter contains the value "even", the function is supposed to return count of how many of first 5 parameters are even.

Code:

```
public class EvenOdd {
    public static int CountNumbers(int num1, type String) {
        int count = 0;
        int[] numbers = {num1, num2, num3, num4, num5};
        if (type.equalsIgnoreCase("even")) {
            for (int number : numbers) {
                if (number % 2 == 0) {
                    count++;
                }
            }
        } else if (type.equalsIgnoreCase("odd")) {
            for (int number : numbers) {
                if (number % 2 != 0) {
                    count++;
                }
            }
        } else {
            .
        }
    }
}
```



```
throw new IllegalArgumentException("The type must be 'even' or 'odd'");
```

```
}
```

```
return count;
```

```
}
```

```
public static void main (String [] args) {
```

```
    java.util.Scanner scanner = new java.util.Scanner(System.in);
```

```
    System.out.println("Enter first numbers");
```

```
    int num1 = scanner.nextInt();
```

```
    System.out.println("Enter second numbers");
```

```
    int num2 = scanner.nextInt();
```

```
    scanner.nextLine();
```

```
    System.out.println("Enter 'even' or 'odd' : ");
```

```
    String type = scanner.nextLine();
```

```
    int result = countNumbers(num1, num2, num3, type);
```

```
    System.out.println("Count : " + result);
```

```
    scanner.close();
```

```
}
```

```
}
```

Input: num1 = 12; Output: 3.

num2 = 17;

num3 = 19;

num4 = 14;

num5 = 115;

type = "odd"

- 1.5 Aim: Write a function that returns the second last digit of the given number. Second last digit is being referred to digit in tens place in given number.

Code:

```
import java.util.Scanner;

public class DigitExtractor {

    public static int getSecondLastDigit(int number) {
        if (Math.abs(number) < 10) {
            throw new IllegalArgumentException("Two digits.");
        }
        int lastTwoDigits = Math.abs(number) % 100;
        int secondLastDigit = lastTwoDigits / 10;
        return secondLastDigit;
    }

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

        try {
            int result = getSecondLastDigit(number);
            System.out.println("Second last Digit: " + result);
        } catch (IllegalArgumentException e) {
            System.out.println(e.getMessage());
        }
        scanner.close();
    }
}
```


Input: 197Output: 7

- 1.6 Aim: Given two strings a and b, print a new string which is made of the following combination - first character of a then, first character of b and so on.

code:

import java.util.Scanner;

public class StringCombiner {

public static String combineStrings(String a, String b) {

StringBuffer result = new StringBuffer();

int lenA = a.length();

int lenB = b.length();

for (int i = 0; i < min(lenA, lenB); i++) {

result.append(a.charAt(i));

result.append(b.charAt(i));

}

if (lenA > lenB) {

result.append(a.substring(lenB));

} else {

result.append(b.substring(lenA));

}

return result.toString();

}

public static void main (String[] args) {

```

Scanner scanner = new Scanner(System.in);
System.out.print("Enter the first String: ");
String a = scanner.nextLine();
System.out.print("Enter the second String: ");
String b = scanner.nextLine();
String result = combineStrings(a, b);
System.out.println("Combined String: " + result);
scanner.close();
}

```

Input: Enter First string : Hello
Enter Second String : World

Output: HelloWorld

- Aim: The padovan sequence is a sequence of numbers that are named after Richard padovan, who attributed its discovery to dutch architect hans van der laan.

Code:

```

import java.util.Scanner;

public class PadovanSequence {
    public static int[] generatePadovanSequence(int n) {
        if (n <= 0) {
            return new int[0];
        }
        int[] sequence = new int[n];
    }
}

```

```

1 // (n2=1) Sequence [1]
2 // (n2=2) Sequence [1, 1]
3 // (n2=3) Sequence [1, 1, 1]
4 // (n2=4) Sequence [1, 1, 1, 1]
5 // (n2=5) Sequence [1, 1, 1, 1, 1]
6 // (n2=6) Sequence [1, 1, 1, 1, 1, 1]
7 // (n2=7) Sequence [1, 1, 1, 1, 1, 1, 1]
8 // (n2=8) Sequence [1, 1, 1, 1, 1, 1, 1, 1]
9 // (n2=9) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1]
10 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

```

```

3 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
4 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
5 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
6 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
7 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
8 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
9 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
10 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

```

```

3 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
4 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
5 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
6 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
7 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
8 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
9 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
10 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

```

```

3 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
4 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
5 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
6 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
7 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
8 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
9 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
10 // (n2=10) Sequence [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

```

Output: Padovan Sequence up to 10: 1 1 1 2 2 3 4 5 7 9.

- 1.8 Aim: Given an array arr of n positive integers, your task is to find all the leaders in the array. An element of the array is a leader if it is greater than or equal to the maximum element on its right side. The rightmost element is always a leader.

Code:

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Scanner;

public class ArrayLeaders {
    public static List<Integer> findLeaders(int[] arr) {
        List<Integer> leaders = new ArrayList<>();
        if (arr == null || arr.length == 0) {
            return leaders;
        }
        int maxFromRight = arr[arr.length - 1];
        leaders.add(maxFromRight);
        for (int i = arr.length - 2; i >= 0; i--) {
            if (arr[i] >= maxFromRight) {
                leaders.add(arr[i]);
                maxFromRight = arr[i];
            }
        }
    }
}
```

```
collections: reverse(leader);
```

```
return leader;
```

```
}
```

```
public static void main (String args[]) {
```

```
    Scanner scanner = new Scanner (System.in);
```

```
    System.out.print("Enter no. of elements in array:");
```

```
    int n = scanner.nextInt();
```

```
    int[] arr = new int[n];
```

```
    System.out.print("Enter elements of array:");
```

```
    for (int i = 0; i < n; i++) {
```

```
        arr[i] = scanner.nextInt();
```

```
    }
```

```
    List<Integer> leaders = findLeaders(arr);
```

```
    System.out.println("Leaders in the array:");
```

```
    for (int leader : leaders) {
```

```
        System.out.print(leader + " ");
```

```
    }
```

```
    System.out.println();
```

```
    scanner.close();
```

```
}
```

```
}
```

Input: n = 6, arr[] = {16, 17, 4, 3, 5, 2}

Output: 17 5 2

1-9

Aim: Given a number 'N' and its reverse, find the number obtained when the number is raised to power of its own reverse.

Code:

```
import java.util.Scanner;

public class powerOfReverse {
    public static int reverseNumber (int number) {
        int reversed = 0;
        while (number != 0) {
            int digit = number % 10;
            reversed = reversed * 10 + digit;
        }
        return reversed;
    }

    public static long powerOfReverse (int num, int rev) {
        return (long) Math.pow(number, reverse);
    }

    public static void main (String args[]) {
        Scanner scanner = new Scanner (System.in);
        System.out.print ("Enter a number: ");
        int number = scanner.nextInt();
        int reverse = reverseNumber (number);
        long result = powerOfReverse (num, rev);
        System.out.println (number + " raised to power " +
            reverse + " is: " + result);
    }
}
```



```
scanner.close();
```

```
}
```

```
}
```

Input: N = 57, R = 75

Output: 262642770.

1.10 Aim: Find the mean of elements of array.

Code:

```
import java.util.Scanner;
```

```
public class MeanCalculator {
```

```
    public static double calculateMean(int[] arr) {
```

```
        if (arr.length == 0) {
```

```
            return 0;
```

```
        }
```

```
        int sum = 0
```

```
        for (int num : arr) {
```

```
            sum += num;
```

```
        }
```

```
        return (double) sum / arr.length;
```

```
    }
```

```
    public static void main(String args[]) {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter no. of elements in array:");
```

```

int n = scanner.nextInt();
int[] arr = new int[n];
System.out.println("Enter the elements of array:");
for (int i = 0; i < n; i++) {
    arr[i] = scanner.nextInt();
}
double mean = calculateMean(arr);
System.out.println("the mean is: %.2f", mean);
scanner.close();
}
}

```

Input: 1 2 3 4 5

Output: 3.0