**JAVA LAB 1**

**Q1. write a program called checkPassFail which prints "PASS" if the int variable "mark" is more than or equal to 50 , or prints "FAIL " otherwise. The program shall always print "DONE" before exiting ?**

import java.util.Scanner;

public class checkPassFail {

public static void main(String[] args) {

Scanner marks = new Scanner(System.in);

System.out.println("Enter marks");

String userMarks = marks.nextLine();

int i = Integer.parseInt(userMarks);

if(i>=50){

System.out.println("PASS");

}

else {

System.out.println("FAIL");

}

System.out.println("DONE");

}

}

**output** : Enter marks : 52 ,

PASS ,

DONE

Enter marks: 49

FAIL

DONE

**Q2**. **write a program called check Odd Even which prints "Odd numbers" if the int variable "number" is odd , or "even number" otherwise. The program shall always print "bye" before existing.**

import java.util.Scanner;

public class checkOddEven {

public static void main(String[] args) {

Scanner number = new Scanner(System.in);

System.out.println("Enter number to check even and odd :");

int i = number.nextInt();

if(i%2!=0){

System.out.println("Odd numbers");

}

else {

System.out.println("Even numbers");

}

System.out.println("bye !");

}

}

Output : Enter number to check even and odd : 12

Even numbers

bye !

Enter number to check even and odd : 13

Odd numbers

bye !

**Q3. write a program called SumAverageRunningInt to produce the sum of 1,2,3,...,to 100 .Store 1 and 100 in variables lowerbound and upperbound ,so that we can change their values easily.Also compare and display the average.**

public class SumAverageRunningInt {

public static void main(String[] args) {

int lowerbound = 1;

int upperbound = 100;

//calculate the sum of integers from lowerbound to upperbound

int sum=0;

for(int i=lowerbound;i<=upperbound;i++){

sum =sum+i;

}

//calculate average

double average = (double) sum/(upperbound-lowerbound+1);

//display

System.out.println("the sum of integers is :"+ lowerbound + "to" + upperbound +"is =" + " " + sum);

System.out.println("The average is :" +average);

}

}

Output : the sum of integers is :1to100is = 5050 The average is :50.5

**Q.4 //write a program called HarmonicSum to compute the sum of a harmonic series , where n=50000**

public class HarmonicSum {

public static void main(String[] args) {

int n=50000;

double sum = 0.0;

for(int i=1;i<=n;i++){

sum = sum+1.0/i;

}

System.out.println("the sum of the harmonic series up to n = " + n + " " +"is:" +" " +sum);

}

}

**Output** : the sum of the harmonic series up to n = 50000 is: 11.397003949278504

**Q.5 write a program called Fibonacci to print the first 20 Fibonacci number F(n) , whereF(n)=F(n-1)+F(n-2) and F(1)=F(2)=1. also compare their average**

public class Fibonacci {

public static void main(String[] args) {

int n = 20;

long[] fibonacciNumbers = new long[n];

//first two fibonacci numbers

fibonacciNumbers[0] = 1;

fibonacciNumbers[1] = 1;

//generate the rest of the fibonacci numbers

for(int i=2;i<n;i++){

fibonacciNumbers[i]=fibonacciNumbers[i-1]+fibonacciNumbers[i-2];

}

//print the fibonacci numbers

System.out.println("The first 20 fibonacci numbers are ");

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

System.out.println(fibonacciNumbers[i]+ "");

}

//calculate and print the average of the fibonacci numbers

long sum=0;

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

sum+=fibonacciNumbers[i];

}

double average = (double) sum/n;

System.out.println("\n The average of the first 20 fibonacci numbers is: "+average);

}

}

**Output :** The first 20 fibonacci numbers are :

1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765

The average of the first 20 fibonacci numbers is: 885.5

**Q.6 write a program called ExtraDigits to extract each digit from an int , in the reverse order, for example if the int is 15423,the output shallbe "3 2 4 5 1" with a space seperating the digits.**

import java.util.Scanner;

public class ExtraDigits {

public static void main(String[] args) {

Scanner digit = new Scanner(System.in);

System.out.println("Enter an integer :");

int n = digit.nextInt();

System.out.println("The digits in reverse order are :");

while(n!=0){

int digits = n%10;

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

n=n/10;

}

} }

**Output** : Enter an integer : 12345

The digits in reverse order are : 5 4 3 2 1

**Q7. write a program that prompts user for a positive integer. The program shall read the input as int and print the "reverse" of the input integer**

import java.util.Scanner;

public class User {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter a positive integer :");

int number = sc.nextInt();

if(number<=0){

System.out.println("Please enter a positive integer");

}

Else

{

int reverseNumber=0;

//reverse the integer

while(number!=0){

int digit=number%10;

reverseNumber=reverseNumber\*10+digit;

number=number/10;

}

System.out.println("the reverse of the entered integer is :" + reverseNumber);

}

} }

**Output** : Enter a positive integer : 12345

the reverse of the entered integer is :54321

**Q.8 Write a Program called TimeTable that prompts user for the size and prints the multiplication Table**

import java.util.Scanner;

public class TimeTable {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the size of the multiplication table: ");

int size = sc.nextInt();

for (int i = 1; i <= size; i++) {

for (int j = 1; j <= size; j++) {

System.out.print(i \* j + "\t");

}

System.out.println();

}

sc.close();

}

}

**Output** : Enter the size of the multiplication table: 2

1 2

2 4

**Q.9 #**

import java.util.Scanner;

public class TriangularPatternA {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt(); // The number of lines to print

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

System.out.print("# ");

}

System.out.println();

}

}

}

**# #**

**# # #**

**# # # #**

**# # # # #**

**# # # # # #**

**# # # # # # #**

**# # # # # # # #**

import java.util.Scanner;

public class TriangularPatternB {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter number of line ");

int n = sc.nextInt();

for(int i=n;i>=1;i--){

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

System.out.print(" # ");

}

System.out.println(" ");

}

}

}

**# # # # # # # #**

**# # # # # # #**

**# # # # # #**

**# # # # #**

**# # # #**

**# # #**

**# #**

**#**

**//# # # # # # # #**

public class TriangularPatternC {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter number of line");

int n = sc.nextInt();

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

// Print leading spaces

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

System.out.print(" ");

}

// Print the # symbols

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

System.out.print("# ");

}

// Move to the next line

System.out.println();

}

}

}

**// # # # # # # #**

**// # # # # # #**

**// # # # # #**

**// # # # #**

**// # # #**

**// # #**

**// #**

public class TriangularPatternD {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter number of lines");

int n = sc.nextInt();

for (int i = 1; i <= n; i++) {

// Print leading spaces

for (int j = n; j > i; j--) {

System.out.print(" ");

}

// Print the # symbols

for (int j = 1; j <= i; j++) {

System.out.print("# ");

}

// Move to the next line

System.out.println();

}}}

**// #**

**// # #**

**// # # #**

**// # # # #**

**// # # # # #**

**// # # # # # #**

**// # # # # # # #**

**//# # # # # # # #**

**Q10. Write a program called Dec2Hex that prompts user for a positive decimal number read as int and print its equivalent hexadecimal string**

import java.util.Scanner;

public class Dec2Hex {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Prompt user for a positive decimal number

System.out.print("Enter a positive decimal number: ");

int decimalNumber = sc.nextInt();

// Check if the input is positive

if (decimalNumber < 0) {

System.out.println("Please enter a positive number.");

} else {

// Convert decimal to hexadecimal

String hexString = Integer.toHexString(decimalNumber).toUpperCase();

// Print the equivalent hexadecimal string

System.out.println("The hexadecimal equivalent is: " + hexString);

}

sc.close();

}

}

**Output** :- Enter a positive decimal number: 255

The hexadecimal equivalent is: FF

**Q.11. Java Program to check whether the given integer is a multiple of both 5 and 7**

import java.util.Scanner;

public class CheckMultiple {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter an Integer : ");

int number = sc.nextInt();

if (number % 5 == 0 && number % 7 == 0) {

System.out.println(number + " is a multiple of both 5 and 7.");

} else {

System.out.println(number + " is not a multiple of both 5 and 7.");

}

sc.close();

}

}

**Output** :- Enter an Integer : 35

35 is a multiple of both 5 and 7.

**Q.12. Java program to check whether the given integer is a prime number or not**

import java.util.Scanner;

public class CheckPrime {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt user for an integer

System.out.print("Enter an integer: ");

int number = scanner.nextInt();

// Check if the number is prime

boolean isPrime = true;

if (number <= 1) {

isPrime = false;

} else {

for (int i = 2; i <= number / 2; i++) {

if (number % i == 0) {

isPrime = false;

break;

}

}

}

// Print the result

if (isPrime) {

System.out.println(number + " is a prime number.");

} else {

System.out.println(number + " is not a prime number.");

}

scanner.close();

}

}

**OutPut :-** Enter an integer: 13

13 is a prime number.

**Q.13. Java program to generate the prime numbers from 1 to N .**

import java.util.Scanner;

public class PrimeNumber {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Prompt user for input

System.out.print("Enter a number (N): ");

int n = sc.nextInt();

System.out.println("Prime numbers between 1 and " + n + ":");

// Check prime numbers from 1 to n

for (int num = 2; num <= n; num++) {

boolean isPrime = true;

// Check if num is a prime number

for (int i = 2; i <= num / 2; i++) {

if (num % i == 0) {

isPrime = false;

break;

}

}

// Print num if it is prime

if (isPrime) {

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

}

}

sc.close();

}

}

**Output** :- Enter a number (N): 100 Prime numbers between 1 and 100: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

**Q14. Java program to find the factorial of a number using recursion**

import java.util.Scanner;

public class Factorial {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt user for input

System.out.print("Enter a non-negative integer: ");

int n = scanner.nextInt();

// Calculate factorial using recursion

long factorial = calculateFactorial(n);

// Print the result

System.out.println("Factorial of " + n + " is: " + factorial);

scanner.close();

}

// Recursive method to calculate factorial

public static long calculateFactorial(int n) {

// Base case: factorial of 0 is 1

if (n == 0) {

return 1;

}

// Recursive case: n \* factorial(n-1)

return n \* calculateFactorial(n - 1);

}

}

**Output** :- Enter a non-negative integer: 5 Factorial of 5 is: 120

**Q15. Java program to implement Binary Search**

import java.util.Scanner;

public class BinarySearch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Prompt user for input

System.out.print("Enter the number of elements in the array: ");

int n = sc.nextInt();

int[] arr = new int[n];

System.out.println("Enter " + n + " sorted integers:");

// Read the sorted array elements

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

arr[i] = sc.nextInt();

}

// Prompt user for the element to search

System.out.print("Enter the element to search: ");

int target = sc.nextInt();

// Perform binary search

int index = binarySearch(arr, target);

// Print the result

if (index != -1) {

System.out.println(target + " found at index " + index);

} else {

System.out.println(target + " not found in the array.");

}

sc.close(); }

// Binary search method

public static int binarySearch(int[] arr, int target) {

int left = 0;

int right = arr.length - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

// Check if target is present at mid

if (arr[mid] == target) {

return mid;

}

// If target is greater, ignore left half

if (arr[mid] < target) {

left = mid + 1;

}

// If target is smaller, ignore right half

else {

right = mid - 1;

}

}

// Element is not present in array

return -1;

}

}

**Output** :- Enter the number of elements in the array: 4

Enter 4 sorted integers:10 11 12 13

Enter the element to search:11

11 found at index 1

**Q16. Java program to find the number of odd numbers in an array**

import java.util.Scanner;

public class OddNumber {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Prompt user for input

System.out.print("Enter the number of elements in the array: ");

int n = sc.nextInt();

int[] arr = new int[n];

System.out.println("Enter " + n + " integers:");

// Read array elements

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

arr[i] = sc.nextInt();

}

// Count odd numbers

int oddCount = countOddNumbers(arr);

// Print the result

System.out.println("Number of odd numbers in the array: " + oddCount);

sc.close();

}

// Method to count odd numbers in an array

public static int countOddNumbers(int[] arr) {

int count = 0;

for (int num : arr) {

if (num % 2 != 0) { // Check if number is odd

count++;

}

}

return count;

}

}

**Output** :- Enter the number of elements in the array: 4

Enter 4 integers:10 12 14 15

Number of odd numbers in the array: 1

**Q17. Java program to check whether a string is a palindrome or not**

public class PalindromeChecker {

public static void main(String[] args) {

String str = "madam"; // You can change this string to test other inputs

boolean isPalindrome = checkPalindrome(str);

if (isPalindrome) {

System.out.println(str + " is a palindrome.");

} else {

System.out.println(str + " is not a palindrome.");

}

}

public static boolean checkPalindrome(String str) {

int length = str.length();

for (int i = 0; i < length / 2; i++) {

if (str.charAt(i) != str.charAt(length - i - 1)) {

return false;

}

}

return true; } }

**Output** :- madam is a palindrome.

**Q.18. Java Program to implement matrix multiplication .**

public class MatrixMultiplication {

public static void main(String[] args) {

// Define two matrices

int[][] matrixA = {

{1, 2, 3},

{4, 5, 6}

};

int[][] matrixB = {

{7, 8},

{9, 10},

{11, 12}

};

// Check if multiplication is possible

if (matrixA[0].length != matrixB.length) {

System.out.println("Matrix multiplication is not possible.");

return;

}

// Perform matrix multiplication

int[][] result = multiplyMatrices(matrixA, matrixB);

// Print the result

printMatrix(result);

}

public static int[][] multiplyMatrices(int[][] matrixA, int[][] matrixB) {

int rowsA = matrixA.length;

int colsA = matrixA[0].length;

int colsB = matrixB[0].length;

int[][] result = new int[rowsA][colsB];

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

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

for (int k = 0; k < colsA; k++) {

result[i][j] += matrixA[i][k] \* matrixB[k][j];

}

}

}

return result;

}

public static void printMatrix(int[][] matrix) {

for (int[] row : matrix) {

for (int element : row) {

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

}

System.out.println();

}

}

}

**Output** :- 58 64

139 154

**Q.19. Java program to check Leap year**

import java.util.Scanner;

public class LeapYear {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a year: ");

int year = scanner.nextInt();

if (isLeapYear(year)) {

System.out.println(year + " is a leap year.");

} else {

System.out.println(year + " is not a leap year.");

}

scanner.close();

}

public static boolean isLeapYear(int year) {

// Leap year is divisible by 4

// but not divisible by 100 unless it is also divisible by 400

if (year % 4 == 0) {

if (year % 100 == 0) {

if (year % 400 == 0) {

return true;

} else {

return false;

}

} else {

return true;

}

} else {

return false;

}

}

}

**Output** :- Enter a year: 2024

2024 is a leap year.

**Q.20. Java program to find the largest and smallest numbers from a list of numbers.**

import java.util.Scanner;

public class LargestSmallestFind {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of elements: ");

int n = scanner.nextInt();

int[] numbers = new int[n];

System.out.println("Enter the numbers:");

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

numbers[i] = scanner.nextInt();

}

int largest = findLargest(numbers);

int smallest = findSmallest(numbers);

System.out.println("Largest number: " + largest);

System.out.println("Smallest number: " + smallest);

scanner.close();

}

public static int findLargest(int[] numbers) {

int largest = numbers[0];

for (int number : numbers) {

if (number > largest) {

largest = number;

}

}

return largest;

}

public static int findSmallest(int[] numbers) {

int smallest = numbers[0];

for (int number : numbers) {

if (number < smallest) {

smallest = number;

}

}

return smallest;

}

}

**Output** :- Enter the number of elements: 5

Enter the numbers: 12 -1 12 15 20

Largest number: 20

Smallest number: -1