1. reverse a word using loop

import java.util.Scanner;

public class ReverseWord {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String word = sc.next();

String reversed = "";

for (int i = word.length() - 1; i >= 0; i--) {

reversed += word.charAt(i);

}

System.out.println("Reversed Word: " + reversed);

sc.close();

}

}

2. check the entered user name is valid or not

import java.util.Scanner;

public class UsernameValidation {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter username: ");

String username1 = sc.next();

System.out.print("Re-enter username: ");

String username2 = sc.next();

if (username1.equals(username2)) {

System.out.println("Username is Valid");

} else {

System.out.println("Username is Invalid");

}

sc.close();

}

}

3.reverse a number using loop?

import java.util.Scanner;

public class ReverseNumber {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

int reversed = 0;

while (num != 0) {

int digit = num % 10;

reversed = reversed \* 10 + digit;

num /= 10;

}

System.out.println("Reversed Number: " + reversed);

sc.close();

}

}

4. eligible for vote or no

import java.util.Scanner;

public class VotingEligibility {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter your age: ");

int age = sc.nextInt();

if (age >= 18) {

System.out.println("You are eligible to vote.");

} else {

System.out.println("You are allowed to vote after " + (18 - age) + " years.");

}

sc.close();

}

}

5. LCM and GCD

import java.util.Scanner;

public class LCM\_GCD {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter first number: ");

int num1 = sc.nextInt();

System.out.print("Enter second number: ");

int num2 = sc.nextInt();

int a = num1, b = num2;

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

int gcd = a;

int lcm = (num1 \* num2) / gcd;

System.out.println("GCD: " + gcd);

System.out.println("LCM: " + lcm);

sc.close(); }}

6.Right Triangle Star Pattern

import java.util.Scanner;

public class RightTrianglePattern {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

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

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

System.out.print("\* ");

}

System.out.println();

}

sc.close();

}

}

7.calculate the simple interest.

import java.util.Scanner;

public class SimpleInterest {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter principal amount: ");

double principal = sc.nextDouble();

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

int years = sc.nextInt();

System.out.print("Is customer a senior citizen (y/n)? ");

char senior = sc.next().charAt(0);

double rate = (senior == 'y' || senior == 'Y') ? 12.0 : 10.0;

double interest = (principal \* rate \* years) / 100;

System.out.println("Interest: " + interest);

sc.close();

}

}

8.Even Sum of Fibonacci Series

import java.util.Scanner;

public class EvenSumFibonacci {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the value of N: ");

int n = sc.nextInt();

int a = 0, b = 1, sum = 0;

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

if (i % 2 == 0) {

sum += a;

}

int next = a + b;

a = b;

b = next;

}

System.out.println("Sum of even index Fibonacci numbers: " + sum);

sc.close();

}

}

9.numbers from M to N by skipping k

import java.util.Scanner;

public class SkipNumbers {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter M: ");

int m = sc.nextInt();

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

int n = sc.nextInt();

System.out.print("Enter K: ");

int k = sc.nextInt();

for (int i = m; i <= n; i += k) {

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

}

sc.close();

}

}

10. matrix addition

import java.util.Scanner;

public class MatrixAddition {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int[][] mat1 = new int[2][2];

int[][] mat2 = new int[2][2];

int[][] sum = new int[2][2];

System.out.println("Enter first matrix:");

for (int i = 0; i < 2; i++)

for (int j = 0; j < 2; j++)

mat1[i][j] = sc.nextInt();

System.out.println("Enter second matrix:");

for (int i = 0; i < 2; i++)

for (int j = 0; j < 2; j++)

mat2[i][j] = sc.nextInt();

for (int i = 0; i < 2; i++)

for (int j = 0; j < 2; j++)

sum[i][j] = mat1[i][j] + mat2[i][j];

System.out.println("Sum of matrices:");

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

for (int j = 0; j < 2; j++) {

System.out.print(sum[i][j] + " ");

}

System.out.println();

}

sc.close();

}

}

11.print rectangle symbol pattern

import java.util.Scanner;

public class RectanglePattern {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the symbol: ");

char symbol = sc.next().charAt(0);

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

int rows = sc.nextInt();

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

int cols = sc.nextInt();

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

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

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

}

System.out.println();

}

sc.close();

}

}

12. sort a list of names in alphabetical order

import java.util.Arrays;

import java.util.Scanner;

public class SortNames {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

String[] names = new String[n];

System.out.println("Enter names:");

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

names[i] = sc.next();

}

System.out.print("Sort Order (A for Ascending, D for Descending): ");

char order = sc.next().charAt(0);

Arrays.sort(names);

if (order == 'D' || order == 'd') {

for (int i = n - 1; i >= 0; i--) {

System.out.println(names[i]);

}

} else {

for (String name : names) {

System.out.println(name);

}

}

sc.close();

}

}

13. matrix multiplication

import java.util.Scanner;

public class MatrixMultiplication {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int[][] mat1 = new int[2][2];

int[][] mat2 = new int[2][2];

int[][] result = new int[2][2];

System.out.println("Enter first matrix:");

for (int i = 0; i < 2; i++)

for (int j = 0; j < 2; j++)

mat1[i][j] = sc.nextInt();

System.out.println("Enter second matrix:");

for (int i = 0; i < 2; i++)

for (int j = 0; j < 2; j++)

mat2[i][j] = sc.nextInt();

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

for (int j = 0; j < 2; j++) {

for (int k = 0; k < 2; k++) {

result[i][j] += mat1[i][k] \* mat2[k][j];

}

} }

System.out.println("Matrix Multiplication Result:");

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

for (int j = 0; j < 2; j++) {

System.out.print(result[i][j] + " ");

}

System.out.println();

}

sc.close();

}

}

14.pyramid pattern

import java.util.Scanner;

public class NumberPyramid {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the number to be printed: ");

int num = sc.nextInt();

System.out.print("Enter max times to print: ");

int max = sc.nextInt();

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

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

System.out.print(num);

}

System.out.println();

}

for (int i = max - 1; i > 0; i--) {

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

System.out.print(num);

}

System.out.println();

}

sc.close();

}

}

15.special characters separately

import java.util.Scanner;

public class SpecialCharactersCount {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String input = sc.nextLine();

int count = 0;

System.out.print("Special Characters: ");

for (char c : input.toCharArray()) {

if (!Character.isLetterOrDigit(c)) {

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

count++;

}

}

System.out.println("\nNumber of Special Characters: " + count);

sc.close();

}

}

16.print all the composite numbers

import java.util.Scanner;

public class CompositeNumbers {

public static boolean isComposite(int num) {

if (num < 4) return false;

for (int i = 2; i <= Math.sqrt(num); i++) {

if (num % i == 0) return true;

}

return false;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter A: ");

int a = sc.nextInt();

System.out.print("Enter B: ");

int b = sc.nextInt();

System.out.print("Composite numbers: ");

for (int i = a; i <= b; i++) {

if (isComposite(i)) {

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

}

}

sc.close();

}

}

17.print the Inverted Full Pyramid

import java.util.Scanner;

public class InvertedFullPyramid {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

for (int i = n; i > 0; i--) {

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

System.out.print(" ");

}

for (int k = 0; k < 2 \* i - 1; k++) {

System.out.print("\*");

}

System.out.println();

}

sc.close();

}

}

18. Mean, Median, Mode of the array of numbers?

import java.util.Arrays;

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class MeanMedianMode {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

int[] arr = new int[n];

System.out.println("Enter elements:");

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

arr[i] = sc.nextInt();

}

double sum = 0;

for (int num : arr) {

sum += num;

}

double mean = sum / n;

Arrays.sort(arr);

double median = (n % 2 == 0) ? (arr[n / 2 - 1] + arr[n / 2]) / 2.0 : arr[n / 2];

Map<Integer, Integer> freqMap = new HashMap<>();

int mode = arr[0], maxCount = 0;

for (int num : arr) {

freqMap.put(num, freqMap.getOrDefault(num, 0) + 1);

if (freqMap.get(num) > maxCount) {

maxCount = freqMap.get(num);

mode = num;

}

}

System.out.println("Mean = " + mean);

System.out.println("Median = " + median);

System.out.println("Mode = " + mode);

sc.close();

}

}

19.Find the factorial of n?

import java.util.Scanner;

public class Factorial {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

long fact = 1;

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

fact \*= i;

}

System.out.println(n + " Factorial = " + fact);

sc.close();

}

}

20.print symbol pyramid pattern

import java.util.Scanner;

public class SymbolPyramid {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the character to be printed: ");

char ch = sc.next().charAt(0);

System.out.print("Enter max number of times printed: ");

int max = sc.nextInt();

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

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

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

}

System.out.println();

}

sc.close();

}

}

21.Find the year of the given date is leap year or not

import java.util.Scanner;

public class LeapYear {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter Year: ");

int year = sc.nextInt();

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {

System.out.println("Given year is a Leap Year");

} else {

System.out.println("Given year is NOT a Leap Year");

}

sc.close();

}

}

22.Find the number of factors for the given number

import java.util.Scanner;

public class FactorsCount {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

int count = 0;

System.out.print("Factors: ");

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

if (num % i == 0) {

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

count++;

}

}

System.out.println("\nNumber of factors: " + count);

sc.close();

}

}

23.given number is Perfect number or not?

import java.util.Scanner;

public class PerfectNumber {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

int sum = 0;

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

if (num % i == 0) {

sum += i;

}

}

if (sum == num) {

System.out.println(num + " is a Perfect Number");

} else {

System.out.println(num + " is NOT a Perfect Number");

}

sc.close();

}

}

24.e number of vowels

import java.util.Scanner;

public class VowelCount {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String str = sc.nextLine();

int count = 0;

str = str.toLowerCase();

for (char c : str.toCharArray()) {

if ("aeiou".indexOf(c) != -1) {

count++;

}

}

System.out.println("Number of vowels: " + count);

sc.close();

}

}

25. print hollow square symbol pattern?

import java.util.Scanner;

public class HollowSquare {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the symbol: ");

char symbol = sc.next().charAt(0);

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

int size = sc.nextInt();

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

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

if (i == 0 || i == size - 1 || j == 0 || j == size - 1) {

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

} else {

System.out.print(" ");

}

}

System.out.println();

}

sc.close();

}

}

26.print consonants and vowels separately

import java.util.Scanner;

public class ConsonantsVowels {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String word = sc.next().toLowerCase();

String vowels = "", consonants = "";

for (char c : word.toCharArray()) {

if ("aeiou".indexOf(c) != -1) {

vowels += c + " ";

} else {

consonants += c + " ";

}

}

System.out.println("Consonants: " + consonants);

System.out.println("Vowels: " + vowels);

sc.close();

}

}

27. find the square, cube

import java.util.Scanner;

public class SquareCube {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

double num = sc.nextDouble();

System.out.println("Square: " + (num \* num));

System.out.println("Cube: " + (num \* num \* num));

sc.close();

}

}

28. find the frequency of each element in the array

import java.util.HashMap;

import java.util.Scanner;

public class FrequencyCount {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

int[] arr = new int[n];

HashMap<Integer, Integer> freq = new HashMap<>();

System.out.println("Enter elements:");

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

arr[i] = sc.nextInt();

freq.put(arr[i], freq.getOrDefault(arr[i], 0) + 1);

}

System.out.println("Element | Frequency");

System.out.println("------------------");

for (int key : freq.keySet()) {

System.out.println(" " + key + " | " + freq.get(key));

}

sc.close();

}

}

29.o find the number of composite numbers in an array

import java.util.Scanner;

public class CountComposites {

public static boolean isComposite(int num) {

if (num < 4) return false;

for (int i = 2; i <= Math.sqrt(num); i++) {

if (num % i == 0) return true;

}

return false;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

int[] arr = new int[n];

int count = 0;

System.out.println("Enter elements:");

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

arr[i] = sc.nextInt();

if (isComposite(arr[i])) {

count++;

}

}

System.out.println("Number of composite numbers: " + count);

sc.close();

}

}

30.Find the nth odd number after n odd number

import java.util.Scanner;

public class NthOddAfterN {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

int nthOdd = (2 \* n) - 1;

int result = nthOdd + (2 \* n);

System.out.println(n + "th odd number after " + n + " odd numbers is: " + result);

sc.close();

}

}

31.t finds whether a given character is present in a string or not

import java.util.Scanner;

public class CharSearch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String str = sc.nextLine();

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

char ch = sc.next().charAt(0);

boolean found = false;

for (int i = 0; i < str.length(); i++) {

if (str.charAt(i) == ch) {

System.out.println(ch + " is found at index: " + i);

found = true;

break;

}

}

if (!found) {

System.out.println(ch + " is not found in the string.");

}

sc.close();

}

}

32.find whether the given number is Armstrong number or not

import java.util.Scanner;

public class ArmstrongNumber {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

int temp = num, sum = 0, digits = String.valueOf(num).length();

while (temp > 0) {

int digit = temp % 10;

sum += Math.pow(digit, digits);

temp /= 10;

}

if (sum == num) {

System.out.println(num + " is an Armstrong number");

} else {

System.out.println(num + " is NOT an Armstrong number");

}

sc.close();

}

}

33. find the sum of digits of N digit number

import java.util.Scanner;

public class SingleDigitSum {

public static int getSingleDigitSum(int num) {

while (num >= 10) {

int sum = 0;

while (num > 0) {

sum += num % 10;

num /= 10;

}

num = sum;

}

return num;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter an N-digit number: ");

int num = sc.nextInt();

System.out.println("Single digit sum: " + getSingleDigitSum(num));

sc.close();

}

}

34.find the square root of a perfect square number

import java.util.Scanner;

public class SquareRoot {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

int sqrt = (int) Math.sqrt(num);

if (sqrt \* sqrt == num) {

System.out.println("Square Root: " + sqrt + ", -" + sqrt);

} else {

System.out.println("Not a perfect square");

}

sc.close();

}

}

35. print inverted pyramid pattern.

import java.util.Scanner;

public class InvertedPyramid {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

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

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

System.out.print(" ");

}

for (int k = 1; k <= (2 \* i - 1); k++) {

System.out.print("\*");

}

System.out.println();

}

sc.close();

}

}

36.count all the prime and composite numbers

import java.util.Scanner;

public class PrimeCompositeCount {

public static boolean isPrime(int num) {

if (num < 2) return false;

for (int i = 2; i <= Math.sqrt(num); i++) {

if (num % i == 0) return false;

}

return true;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

int primeCount = 0, compositeCount = 0;

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

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

int num = sc.nextInt();

if (isPrime(num)) {

primeCount++;

} else if (num > 1) {

compositeCount++;

}

}

System.out.println("Prime numbers: " + primeCount);

System.out.println("Composite numbers: " + compositeCount);

sc.close();

}

}

37.Mth maximum number and Nth minimum number in an array

import java.util.Arrays;

import java.util.Scanner;

public class MthMaxNthMin {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

int[] arr = new int[n];

System.out.println("Enter elements:");

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

arr[i] = sc.nextInt();

}

System.out.print("Enter M (max position): ");

int m = sc.nextInt();

System.out.print("Enter N (min position): ");

int nth = sc.nextInt();

Arrays.sort(arr);

int mthMax = arr[n - m];

int nthMin = arr[nth - 1];

System.out.println(m + "th Maximum Number = " + mthMax);

System.out.println(nth + "th Minimum Number = " + nthMin);

System.out.println("Sum = " + (mthMax + nthMin));

System.out.println("Difference = " + (mthMax - nthMin));

sc.close();

}

}

38. total amount available in the ATM machine

import java.util.Scanner;

public class ATMBalance {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int[] denominations = {2000, 500, 200, 100};

int totalBalance = 0;

for (int i = 0; i < 4; i++) {

System.out.print("Enter denomination " + denominations[i] + " count: ");

int count = sc.nextInt();

totalBalance += count \* denominations[i];

}

System.out.println("Total Available Balance in ATM: " + totalBalance);

sc.close();

}

}

39.check string or number is a palindrome

import java.util.Scanner;

public class PalindromeCheck {

public static boolean isPalindrome(String str) {

int left = 0, 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) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter 1 for String, 2 for Number: ");

int choice = sc.nextInt();

sc.nextLine();

System.out.print("Enter value: ");

String value = sc.nextLine();

if (isPalindrome(value)) {

System.out.println("It is a Palindrome");

} else {

System.out.println("Not a Palindrome");

}

sc.close();

}

}

40.convert Decimal number equivalent

import java.util.Scanner;

public class DecimalConversion {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

System.out.println("Binary: " + Integer.toBinaryString(num));

System.out.println("Octal: " + Integer.toOctalString(num));

sc.close();

}

}

41.o give bonus to all the employees o

import java.util.Scanner;

public class EmployeeBonus {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter employee grade (A/B): ");

char grade = sc.next().charAt(0);

System.out.print("Enter employee salary: ");

double salary = sc.nextDouble();

double bonus = (grade == 'A') ? salary \* 0.05 : salary \* 0.10;

if (salary < 10000) {

bonus += salary \* 0.02;

}

System.out.println("Salary: " + salary);

System.out.println("Bonus: " + bonus);

System.out.println("Total Salary: " + (salary + bonus));

sc.close();

}

}

42.print the first n perfect numbers

import java.util.Scanner;

public class PerfectNumbers {

public static boolean isPerfect(int num) {

int sum = 0;

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

if (num % i == 0) sum += i;

}

return sum == num;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

int count = 0, num = 1;

System.out.print("First " + n + " perfect numbers: ");

while (count < n) {

if (isPerfect(num)) {

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

count++;

}

num++;

}

sc.close();

}

}

43.Then calculate the total and aggregate, display the grade obtained by the student

import java.util.Scanner;

public class StudentGrade {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int subjects = 4;

int[] marks = new int[subjects];

int total = 0;

System.out.println("Enter marks for " + subjects + " subjects:");

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

marks[i] = sc.nextInt();

total += marks[i];

}

double aggregate = total / (double) subjects;

System.out.println("Total = " + total);

System.out.println("Aggregate = " + aggregate + "%");

if (aggregate >= 75) System.out.println("Grade: DISTINCTION");

else if (aggregate >= 60) System.out.println("Grade: FIRST DIVISION");

else if (aggregate >= 50) System.out.println("Grade: SECOND DIVISION");

else if (aggregate >= 40) System.out.println("Grade: THIRD DIVISION");

else System.out.println("Grade: FAIL");

sc.close();

}

}

44.o calculate tax

import java.util.Scanner;

public class TaxCalculator {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter your income: ");

double income = sc.nextDouble();

double tax = 0;

if (income > 500000) tax = income \* 0.30;

else if (income > 300000) tax = income \* 0.20;

else if (income > 150000) tax = income \* 0.10;

System.out.println("Tax = " + tax);

sc.close();

}

}

45. print the multiplication table of number m up to n.

import java.util.Scanner;

public class MultiplicationTable {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter M: ");

int m = sc.nextInt();

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

int n = sc.nextInt();

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

System.out.println(i + " x " + m + " = " + (i \* m));

}

sc.close();

}

}

46.find average of positive and negative num

import java.util.Scanner;

public class AverageCalculator {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int sumPositive = 0, sumNegative = 0, countPositive = 0, countNegative = 0;

System.out.println("Enter numbers (-1 to stop):");

while (true) {

int num = sc.nextInt();

if (num == -1) break;

if (num > 0) {

sumPositive += num;

countPositive++;

} else {

sumNegative += num;

countNegative++;

}

}

if (countPositive > 0) System.out.println("Average of Positive Numbers: " + (sumPositive / (double) countPositive));

if (countNegative > 0) System.out.println("Average of Negative Numbers: " + (sumNegative / (double) countNegative));

sc.close();

}

}

47.count upper and lower case and numbers util \* is entered

import java.util.Scanner;

public class CharacterCount {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int upper = 0, lower = 0, digits = 0;

System.out.println("Enter characters (\* to stop):");

while (true) {

char ch = sc.next().charAt(0);

if (ch == '\*') break;

if (Character.isUpperCase(ch)) upper++;

else if (Character.isLowerCase(ch)) lower++;

else if (Character.isDigit(ch)) digits++;

}

System.out.println("Uppercase: " + upper);

System.out.println("Lowercase: " + lower);

System.out.println("Digits: " + digits);

sc.close();

}

}

48.Find the Nth Largest Number in a array.

import java.util.Arrays;

import java.util.Scanner;

public class NthLargest {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

int[] arr = new int[n];

System.out.println("Enter elements:");

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

arr[i] = sc.nextInt();

}

System.out.print("Enter N (for Nth largest): ");

int nth = sc.nextInt();

Arrays.sort(arr);

System.out.println(nth + "th largest number: " + arr[n - nth]);

sc.close();

}

}

49.o Remove the Duplicate Items from a array.

import java.util.Arrays;

import java.util.LinkedHashSet;

import java.util.Scanner;

public class RemoveDuplicates {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

int[] arr = new int[n];

System.out.println("Enter elements:");

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

arr[i] = sc.nextInt();

}

LinkedHashSet<Integer> uniqueSet = new LinkedHashSet<>();

for (int num : arr) {

uniqueSet.add(num);

}

System.out.println("Array without duplicates: " + uniqueSet);

sc.close();

}

}

50.e rate of interest Polymorphism

class Bank {

double getRateOfInterest() {

return 0;

}

}

class SBI extends Bank {

double getRateOfInterest() {

return 8.4;

}

}

class ICICI extends Bank {

double getRateOfInterest() {

return 7.3;

}

}

class AXIS extends Bank {

double getRateOfInterest() {

return 9.7;

}

}

public class BankInterest {

public static void main(String[] args) {

Bank sbi = new SBI();

Bank icici = new ICICI();

Bank axis = new AXIS();

System.out.println("SBI Interest Rate: " + sbi.getRateOfInterest() + "%");

System.out.println("ICICI Interest Rate: " + icici.getRateOfInterest() + "%");

System.out.println("AXIS Interest Rate: " + axis.getRateOfInterest() + "%");

}

}

51.member names of a subclass hide members

class SuperClass {

int num;

SuperClass(int num) {

this.num = num;

}

void display() {

System.out.println("SuperClass number: " + num);

}

}

class SubClass extends SuperClass {

int num;

SubClass(int superNum, int subNum) {

super(superNum);

this.num = subNum;

}

void display() {

super.display();

System.out.println("SubClass number: " + num);

}

}

public class SuperKeywordExample {

public static void main(String[] args) {

SubClass obj = new SubClass(100, 200);

obj.display();

}

}

52.Runnable interface in Java to generate Fibonacci series

class FibonacciRunnable implements Runnable {

int n;

FibonacciRunnable(int n) {

this.n = n;

}

public void run() {

int a = 0, b = 1;

System.out.print("Fibonacci Series: " + a + " " + b);

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

int next = a + b;

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

a = b;

b = next;

}

}

}

public class FibonacciThread {

public static void main(String[] args) {

Thread thread = new Thread(new FibonacciRunnable(5));

thread.start();

}

}

53.find the sum of N numbers using array and throw

import java.util.Scanner;

public class ArraySum {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

int[] arr = new int[n];

int sum = 0;

System.out.println("Enter elements:");

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

arr[i] = sc.nextInt();

}

try {

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

sum += arr[i];

}

} catch (ArrayIndexOutOfBoundsException e) {

System.out.println("Error: Index out of bounds!");

}

System.out.println("Sum = " + sum);

sc.close();

}

}

54. Runnable interface in Java to find whether a given number is prime or not

class PrimeCheckRunnable implements Runnable {

int num;

PrimeCheckRunnable(int num) {

this.num = num;

}

public void run() {

boolean isPrime = num > 1;

for (int i = 2; i <= Math.sqrt(num); i++) {

if (num % i == 0) {

isPrime = false;

break;

}

}

System.out.println(num + " is " + (isPrime ? "Prime" : "Not Prime"));

}

}

public class PrimeThread {

public static void main(String[] args) {

Thread thread = new Thread(new PrimeCheckRunnable(5));

thread.start();

}

}

55.o Convert a Given Number of Days in Terms of Years

import java.util.Scanner;

public class DaysConversion {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int days = sc.nextInt();

int years = days / 365;

int weeks = (days % 365) / 7;

int remainingDays = (days % 365) % 7;

System.out.println("Years: " + years);

System.out.println("Weeks: " + weeks);

System.out.println("Days: " + remainingDays);

sc.close();

}

}

56.find the number of student users in the college

import java.util.Scanner;

public class StudentUsers {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter total users: ");

int totalUsers = sc.nextInt();

System.out.print("Enter staff users: ");

int staffUsers = sc.nextInt();

int nonTeachingStaff = staffUsers / 3;

int studentUsers = totalUsers - (staffUsers + nonTeachingStaff);

System.out.println("Student Users: " + studentUsers);

sc.close();

}

}

57.o print number of factors and to print nth factor of the given number

import java.util.ArrayList;

import java.util.Scanner;

public class Factors {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

System.out.print("Enter N (for Nth factor): ");

int n = sc.nextInt();

ArrayList<Integer> factors = new ArrayList<>();

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

if (num % i == 0) {

factors.add(i);

}

}

System.out.println("Number of factors: " + factors.size());

if (n <= factors.size()) {

System.out.println(n + "th Factor: " + factors.get(n - 1));

} else {

System.out.println("Nth factor does not exist.");

}

sc.close();

}

}

58.o print n prime numbers after nth Prime number

import java.util.Scanner;

public class NextPrimes {

public static boolean isPrime(int num) {

if (num < 2) return false;

for (int i = 2; i <= Math.sqrt(num); i++) {

if (num % i == 0) return false;

}

return true;

}

public static int findNthPrime(int n) {

int count = 0, num = 2;

while (true) {

if (isPrime(num)) {

count++;

if (count == n) return num;

}

num++;

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

int nthPrime = findNthPrime(n);

System.out.println(n + "th Prime number: " + nthPrime);

System.out.print(n + " prime numbers after " + nthPrime + ": ");

int count = 0, nextNum = nthPrime + 1;

while (count < n) {

if (isPrime(nextNum)) {

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

count++;

}

nextNum++;

}

sc.close();

}

}

59. create a list of all numbers in a range which are perfect squares a

import java.util.Scanner;

public class PerfectSquares {

public static int digitSum(int num) {

int sum = 0;

while (num > 0) {

sum += num % 10;

num /= 10;

}

return sum;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter lower range: ");

int lower = sc.nextInt();

System.out.print("Enter upper range: ");

int upper = sc.nextInt();

System.out.print("Perfect squares with digit sum < 10: ");

for (int i = lower; i <= upper; i++) {

int sqrt = (int) Math.sqrt(i);

if (sqrt \* sqrt == i && digitSum(i) < 10) {

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

}

}

sc.close();

}

}

60.o print unique permutations of a given number

import java.util.HashSet;

import java.util.Scanner;

import java.util.Set;

public class UniquePermutations {

public static void generatePermutations(String str, String ans, Set<String> result) {

if (str.length() == 0) {

result.add(ans);

return;

}

for (int i = 0; i < str.length(); i++) {

char ch = str.charAt(i);

String rest = str.substring(0, i) + str.substring(i + 1);

generatePermutations(rest, ans + ch, result);

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String num = sc.next();

Set<String> result = new HashSet<>();

generatePermutations(num, "", result);

System.out.println("Unique Permutations: " + result);

sc.close();

}

}

61.a Program to create an array with the First Element as the Number and Second Element as the Square of the Number.

import java.util.Scanner;

public class NumberSquareArray {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter lower range: ");

int lower = sc.nextInt();

System.out.print("Enter upper range: ");

int upper = sc.nextInt();

System.out.println("Generated Array:");

for (int i = lower; i <= upper; i++) {

System.out.println("(" + i + ", " + (i \* i) + ")");

}

sc.close();

}

}

62.n a class to represent a bank account.

import java.util.Scanner;

class BankAccount {

String name;

int accountNumber;

String accountType;

double balance;

BankAccount(String name, int accountNumber, String accountType, double balance) {

this.name = name;

this.accountNumber = accountNumber;

this.accountType = accountType;

this.balance = Math.max(balance, 500);

}

void deposit(double amount) {

balance += amount;

System.out.println("Deposited: " + amount + " | New Balance: " + balance);

}

void withdraw(double amount) {

if (balance - amount >= 500) {

balance -= amount;

System.out.println("Withdrawn: " + amount + " | Remaining Balance: " + balance);

} else {

System.out.println("Insufficient balance! Minimum Rs.500 required.");

}

}

void displayBalance() {

System.out.println("Account Balance: " + balance);

}

}

public class BankSystem {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

BankAccount acc = new BankAccount("Raja", 101, "Savings", 10000);

acc.deposit(5000);

acc.withdraw(3000);

acc.displayBalance();

sc.close(); }

}

63.o Reverse and Add a Number until you get a Palindrome.

import java.util.Scanner;

public class ReverseAddPalindrome {

public static int reverseNumber(int num) {

int reversed = 0;

while (num > 0) {

reversed = reversed \* 10 + num % 10;

num /= 10;

}

return reversed;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

while (true) {

int reverse = reverseNumber(num);

int sum = num + reverse;

System.out.println(num + " + " + reverse + " = " + sum);

num = sum;

if (num == reverseNumber(num)) {

System.out.println("Palindrome found: " + num);

break;

}

}

sc.close();

}

}

64. Customer class with deposit() and withdraw() as synchronized methods

class Bank {

private double balance = 10000;

public synchronized void withdraw(double amount) {

while (balance < amount) {

System.out.println("Insufficient balance. Waiting for deposit...");

try {

wait();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

balance -= amount;

System.out.println("Withdraw successful! Remaining balance: " + balance);

}

public synchronized void deposit(double amount) {

balance += amount;

System.out.println("Deposit successful! New balance: " + balance);

notify();

}

}

public class BankThread {

public static void main(String[] args) {

Bank bank = new Bank();

new Thread(() -> bank.withdraw(12000)).start();

new Thread(() -> bank.deposit(3000)).start();

}

}

65."FizzBuzz

import java.util.Scanner;

public class FizzBuzz {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter n: ");

int n = sc.nextInt();

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

if (i % 3 == 0 && i % 5 == 0) System.out.println("FizzBuzz");

else if (i % 3 == 0) System.out.println("Fizz");

else if (i % 5 == 0) System.out.println("Buzz");

else System.out.println(i);

}

sc.close();

}

}

66.o find the common elements in two array of Positive integer

import java.util.HashSet;

import java.util.Scanner;

public class CommonElements {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter size of first array: ");

int n1 = sc.nextInt();

int[] arr1 = new int[n1];

System.out.println("Enter first array elements:");

for (int i = 0; i < n1; i++) arr1[i] = sc.nextInt();

System.out.print("Enter size of second array: ");

int n2 = sc.nextInt();

int[] arr2 = new int[n2];

System.out.println("Enter second array elements:");

for (int i = 0; i < n2; i++) arr2[i] = sc.nextInt();

HashSet<Integer> set = new HashSet<>();

for (int num : arr1) set.add(num);

System.out.print("Common Elements: ");

for (int num : arr2) {

if (set.contains(num)) {

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

}

}

sc.close();

}

}

67.Roman numerals to int

import java.util.HashMap;

import java.util.Scanner;

public class RomanToInteger {

public static int romanToInt(String s) {

HashMap<Character, Integer> map = new HashMap<>();

map.put('I', 1);

map.put('V', 5);

map.put('X', 10);

map.put('L', 50);

map.put('C', 100);

map.put('D', 500);

map.put('M', 1000);

int sum = 0;

for (int i = 0; i < s.length(); i++) {

if (i > 0 && map.get(s.charAt(i)) > map.get(s.charAt(i - 1))) {

sum += map.get(s.charAt(i)) - 2 \* map.get(s.charAt(i - 1));

} else {

sum += map.get(s.charAt(i));

}

}

return sum;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter Roman numeral: ");

String roman = sc.next();

System.out.println("Integer value: " + romanToInt(roman));

sc.close();

}

}

68.n two strings ransomNote and magazine, return true if ransomNote

import java.util.HashMap;

import java.util.Scanner;

public class RansomNote {

public static boolean canConstruct(String ransomNote, String magazine) {

HashMap<Character, Integer> magazineLetters = new HashMap<>();

for (char ch : magazine.toCharArray()) {

magazineLetters.put(ch, magazineLetters.getOrDefault(ch, 0) + 1);

}

for (char ch : ransomNote.toCharArray()) {

if (!magazineLetters.containsKey(ch) || magazineLetters.get(ch) == 0) {

return false;

}

magazineLetters.put(ch, magazineLetters.get(ch) - 1);

}

return true;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter ransom note: ");

String ransomNote = sc.next();

System.out.print("Enter magazine: ");

String magazine = sc.next();

System.out.println("Can construct? " + canConstruct(ransomNote, magazine));

sc.close();

}

}

69.n integer num, return the number of steps to reduce it to zero.

import java.util.Scanner;

public class ReduceToZero {

public static int countSteps(int num) {

int steps = 0;

while (num > 0) {

num = (num % 2 == 0) ? num / 2 : num - 1;

steps++;

}

return steps;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

System.out.println("Steps to reduce to zero: " + countSteps(num));

sc.close();

}

}

70.Develop a programme that uses Multiple Inheritance

interface AcademicMarks {

int[] getMarks();

}

class Student implements AcademicMarks {

String name;

int[] marks = new int[6];

Student(String name, int[] marks) {

this.name = name;

this.marks = marks;

}

public int[] getMarks() {

return marks;

}

double calculateAggregate() {

int total = 0;

for (int mark : marks) {

total += mark;

}

return total / 6.0;

}

String getGrade() {

double aggregate = calculateAggregate();

if (aggregate >= 75) return "DISTINCTION";

else if (aggregate >= 60) return "FIRST DIVISION";

else if (aggregate >= 50) return "SECOND DIVISION";

else if (aggregate >= 40) return "THIRD DIVISION";

else return "FAIL";

}

}

public class StudentGrades {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter student name: ");

String name = sc.next();

int[] marks = new int[6];

System.out.println("Enter marks for 6 subjects:");

for (int i = 0; i < 6; i++) {

marks[i] = sc.nextInt();

}

Student student = new Student(name, marks);

System.out.println("Total Marks: " + student.calculateAggregate() \* 6);

System.out.println("Aggregate: " + student.calculateAggregate() + "%");

System.out.println("Grade: " + student.getGrade());

sc.close();

}

}