1. Reverse of word

public class ReverseWord {

public static void main(String[] args) {

String word = "Hello";

String reversedWord = "";

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

reversedWord += word.charAt(i);

}

System.out.println("Original word: " + word);

System.out.println("Reversed word: " + reversedWord);

}

}

2.user name is valid or not

public class UsernameValidator {

public static void main(String[] args) {

String username = "exampleUser123";

if (isValidUsername(username)) {

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

} else {

System.out.println("Username is not valid.");

}

}

public static boolean isValidUsername(String username) {

// Check length

if (username.length() < 6 || username.length() > 20) {

return false;

}

// Check if it starts with a letter

if (!Character.isLetter(username.charAt(0))) {

return false;

}

// Check if it contains only alphanumeric characters

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

char c = username.charAt(i);

if (!Character.isLetterOrDigit(c)) {

return false;

}

}

return true;

}

}

3.reverse number using loop

public class ReverseNumber {

public static void main(String[] args) {

int number = 12345;

int reversedNumber = 0;

while (number != 0) {

int digit = number % 10; // Get the last digit

reversedNumber = reversedNumber \* 10 + digit; // Append the digit to the reversed number

number /= 10; // Remove the last digit from the number

}

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

}

}

4.eligible to vote or not

import java.util.Scanner;

public class AgeCheck {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int age = scanner.nextInt();

if (age > 18) {

System.out.println("Person is eligible to vote.");

} else {

System.out.println("Person is not eligible to vote.");

}

scanner.close();

}

}

5.lcm and gcd

public class LCMAndGCD {

public static void main(String[] args) {

int num1 = 56;

int num2 = 98;

int gcd = findGCD(num1, num2);

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

System.out.println("GCD of " + num1 + " and " + num2 + " is: " + gcd);

System.out.println("LCM of " + num1 + " and " + num2 + " is: " + lcm);

}

public static int findGCD(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

}

6.right triangle

public class RightTriangle {

public static void main(String[] args) {

int n = 5; // Number of rows

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

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

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

}

System.out.println();

}

}

}

7.simple interest

import java.util.Scanner;

public class SimpleInterest {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

double principal = scanner.nextDouble();

System.out.print("Enter the time period (in years): ");

int time = scanner.nextInt();

System.out.print("Is the person a senior? (yes/no): ");

String isSenior = scanner.next();

double rate;

if (isSenior.equalsIgnoreCase("yes")) {

rate = 12.0; // Rate for seniors

} else {

rate = 10.0; // Rate for others

}

double simpleInterest = (principal \* rate \* time) / 100;

System.out.println("The simple interest is: " + simpleInterest);

scanner.close();

}

}

8.even sum of Fibonacci series

public class EvenFibonacciSum {

public static void main(String[] args) {

int limit = 4000000; // Set a limit to stop the series

int num1 = 1, num2 = 2;

int sum = 0;

while (num1 <= limit) {

if (num1 % 2 == 0) {

sum += num1;

}

// Move to the next number in the series

int nextNum = num1 + num2;

num1 = num2;

num2 = nextNum;

}

System.out.println("Sum of even Fibonacci numbers up to " + limit + " is: " + sum);

}

}

9.print numbers from m to n by skipping k numbers between

public class SkipNumbers {

public static void main(String[] args) {

int m = 1; // Starting number

int n = 20; // Ending number

int k = 2; // Numbers to skip

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

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

}

}

}

10.matrix addition

public class MatrixAddition {

public static void main(String[] args) {

// Define the dimensions of the matrices

int rows = 2;

int columns = 3;

// Initialize the first matrix

int[][] matrix1 = {

{1, 2, 3},

{4, 5, 6}

};

// Initialize the second matrix

int[][] matrix2 = {

{7, 8, 9},

{10, 11, 12}

};

// Create a result matrix to store the sum

int[][] result = new int[rows][columns];

// Perform matrix addition

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

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

result[i][j] = matrix1[i][j] + matrix2[i][j];

}

}

// Print the result matrix

System.out.println("Resultant Matrix:");

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

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

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

}

System.out.println();

}

}

}

11.rectangle pattern

import java.util.Scanner;

public class RectanglePattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

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

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

int rows = scanner.nextInt();

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

int columns = scanner.nextInt();

// Print the rectangle pattern

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

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

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

}

System.out.println();

}

scanner.close();

}

}

12.sort list of names in ascending or descending by getting input from user

import java.util.Arrays;

import java.util.Scanner;

public class SortNames {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input number of names

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

int numberOfNames = scanner.nextInt();

scanner.nextLine(); // Consume the newline

// Input names

String[] names = new String[numberOfNames];

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

System.out.print("Enter name " + (i + 1) + ": ");

names[i] = scanner.nextLine();

}

// Input sort order

System.out.print("Enter sort order (asc/desc): ");

String sortOrder = scanner.nextLine().toLowerCase();

// Sort names

Arrays.sort(names);

if (sortOrder.equals("desc")) {

// Reverse the order for descending sort

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

String temp = names[i];

names[i] = names[names.length - 1 - i];

names[names.length - 1 - i] = temp;

}

}

// Print sorted names

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

for (String name : names) {

System.out.println(name);

}

scanner.close();

}

}

13.matrix multiplication

import java.util.Scanner;

public class MatrixMultiplication {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input dimensions of the first matrix

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

int rows1 = scanner.nextInt();

System.out.print("Enter the number of columns in the first matrix: ");

int columns1 = scanner.nextInt();

// Input dimensions of the second matrix

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

int rows2 = scanner.nextInt();

System.out.print("Enter the number of columns in the second matrix: ");

int columns2 = scanner.nextInt();

// Ensure the matrices can be multiplied

if (columns1 != rows2) {

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

return;

}

// Initialize matrices

int[][] matrix1 = new int[rows1][columns1];

int[][] matrix2 = new int[rows2][columns2];

int[][] result = new int[rows1][columns2];

// Input elements of the first matrix

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

for (int i = 0; i < rows1; i++) {

for (int j = 0; j < columns1; j++) {

matrix1[i][j] = scanner.nextInt();

}

}

// Input elements of the second matrix

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

for (int i = 0; i < rows2; i++) {

for (int j = 0; j < columns2; j++) {

matrix2[i][j] = scanner.nextInt();

}

}

// Perform matrix multiplication

for (int i = 0; i < rows1; i++) {

for (int j = 0; j < columns2; j++) {

for (int k = 0; k < columns1; k++) {

result[i][j] += matrix1[i][k] \* matrix2[k][j];

}

}

}

// Print the resulting matrix

System.out.println("Resulting matrix:");

for (int i = 0; i < rows1; i++) {

for (int j = 0; j < columns2; j++) {

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

}

System.out.println();

}

scanner.close();

}

}

14.print pattern

1

11

111

11

1 by getting the max number of time printed

import java.util.Scanner;

public class PrintPattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the maximum number of times to print '1': ");

int maxTimes = scanner.nextInt();

// Print the first part of the pattern

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

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

System.out.print("1");

}

System.out.println();

}

// Print the second part of the pattern

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

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

System.out.print("1");

}

System.out.println();

}

scanner.close();

}

}

15.special character separately and number of special character in a line

import java.util.Scanner;

public class SpecialCharacterCounter {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a line of text: ");

String line = scanner.nextLine();

StringBuilder specialCharacters = new StringBuilder();

int specialCharacterCount = 0;

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

char c = line.charAt(i);

if (!Character.isLetterOrDigit(c) && !Character.isWhitespace(c)) {

specialCharacters.append(c);

specialCharacterCount++;

}

}

System.out.println("Special characters: " + specialCharacters.toString());

System.out.println("Number of special characters: " + specialCharacterCount);

scanner.close();

}

}

16.composite number between a and b

import java.util.Scanner;

public class CompositeNumbers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int a = scanner.nextInt();

System.out.print("Enter the ending number (b): ");

int b = scanner.nextInt();

System.out.println("Composite numbers between " + a + " and " + b + ":");

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

if (isComposite(i)) {

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

}

}

scanner.close();

}

public static boolean isComposite(int num) {

if (num <= 1) {

return false;

}

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

if (num % i == 0) {

return true;

}

}

return false;

}

}

17.inverted full pyramid

import java.util.Scanner;

public class InvertedPyramid {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int rows = scanner.nextInt();

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

for (int j = rows - i; j > 0; j--) {

System.out.print(" ");

}

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

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

}

System.out.println();

}

scanner.close();

}

}

18.mean, median and mode of array of numbers

import java.util.Arrays;

import java.util.HashMap;

import java.util.Map;

public class Statistics {

public static void main(String[] args) {

int[] numbers = {1, 2, 2, 3, 4, 5, 5, 5, 6, 7};

double mean = calculateMean(numbers);

double median = calculateMedian(numbers);

int mode = calculateMode(numbers);

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

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

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

}

public static double calculateMean(int[] numbers) {

double sum = 0;

for (int num : numbers) {

sum += num;

}

return sum / numbers.length;

}

public static double calculateMedian(int[] numbers) {

Arrays.sort(numbers);

int middle = numbers.length / 2;

if (numbers.length % 2 == 0) {

return (numbers[middle - 1] + numbers[middle]) / 2.0;

} else {

return numbers[middle];

}

}

public static int calculateMode(int[] numbers) {

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

for (int num : numbers) {

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

}

int mode = numbers[0];

int maxCount = 0;

for (Map.Entry<Integer, Integer> entry : frequencyMap.entrySet()) {

if (entry.getValue() > maxCount) {

maxCount = entry.getValue();

mode = entry.getKey();

}

}

return mode;

}

}

19.factorial of n

import java.util.Scanner;

public class Factorial {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

long factorial = calculateFactorial(n);

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

scanner.close();

}

public static long calculateFactorial(int n) {

long result = 1;

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

result \*= i;

}

return result;

}

}

20.print pattern

%

%%

%%%

%%

% by getting symbol and max number of imes printed from user

import java.util.Scanner;

public class CustomPattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

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

System.out.print("Enter the maximum number of times to print the symbol: ");

int maxTimes = scanner.nextInt();

// Print the first part of the pattern

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

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

System.out.print(symbol);

}

System.out.println();

}

// Print the second part of the pattern

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

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

System.out.print(symbol);

}

System.out.println();

}

scanner.close();

}

}

21. leap year or not

import java.util.Scanner;

public class LeapYearCheck {

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) {

if (year % 4 == 0) {

if (year % 100 == 0) {

return year % 400 == 0;

} else {

return true;

}

} else {

return false;

}

}

}

22.number of factors for the given number

import java.util.Scanner;

public class NumberOfFactors {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

int factorCount = calculateFactors(number);

System.out.println("The number of factors for " + number + " is: " + factorCount);

scanner.close();

}

public static int calculateFactors(int number) {

int count = 0;

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

if (number % i == 0) {

count++;

}

}

return count;

}

}

23.perfect number or not

import java.util.Scanner;

public class PerfectNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

if (isPerfectNumber(number)) {

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

} else {

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

}

scanner.close();

}

public static boolean isPerfectNumber(int number) {

int sum = 0;

for (int i = 1; i <= number / 2; i++) {

if (number % i == 0) {

sum += i;

}

}

return sum == number;

}

}

24.number of vowels in a statement

import java.util.Scanner;

public class VowelCounter {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String statement = scanner.nextLine();

int vowelCount = countVowels(statement);

System.out.println("The number of vowels in the statement is: " + vowelCount);

scanner.close();

}

public static int countVowels(String statement) {

int count = 0;

statement = statement.toLowerCase(); // Convert to lowercase for easier comparison

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

char c = statement.charAt(i);

if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {

count++;

}

}

return count;

}

}

25.hollow square by getting symbol from input

import java.util.Scanner;

public class HollowSquare {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

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

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

int size = scanner.nextInt();

// Print the hollow square pattern

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

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

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

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

} else {

System.out.print(" ");

}

}

System.out.println();

}

scanner.close();

}

}

26.consonants and vowels separately in a given word

import java.util.Scanner;

public class SeparateConsonantsAndVowels {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String word = scanner.nextLine();

StringBuilder vowels = new StringBuilder();

StringBuilder consonants = new StringBuilder();

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

char c = word.charAt(i);

if (isVowel(c)) {

vowels.append(c);

} else if (Character.isLetter(c)) {

consonants.append(c);

}

}

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

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

scanner.close();

}

public static boolean isVowel(char c) {

c = Character.toLowerCase(c);

return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u';

}

}

27.square,cube of the decimal number

import java.util.Scanner;

public class SquareAndCube {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

double number = scanner.nextDouble();

double square = Math.pow(number, 2);

double cube = Math.pow(number, 3);

System.out.println("Square of " + number + " is: " + square);

System.out.println("Cube of " + number + " is: " + cube);

scanner.close();

}

}

28.frequency of each element in the array

import java.util.HashMap;

import java.util.Map;

public class ElementFrequency {

public static void main(String[] args) {

int[] numbers = {1, 2, 2, 3, 4, 4, 4, 5};

Map<Integer, Integer> frequencyMap = calculateFrequency(numbers);

System.out.println("Element frequencies:");

for (Map.Entry<Integer, Integer> entry : frequencyMap.entrySet()) {

System.out.println("Element " + entry.getKey() + " occurs " + entry.getValue() + " times.");

}

}

public static Map<Integer, Integer> calculateFrequency(int[] numbers) {

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

for (int num : numbers) {

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

}

return frequencyMap;

}

}

29.number of composite numbers in an array

public class CompositeNumberCount {

public static void main(String[] args) {

int[] numbers = {4, 6, 7, 9, 10, 11, 13, 15};

int compositeCount = countCompositeNumbers(numbers);

System.out.println("The number of composite numbers in the array is: " + compositeCount);

}

public static boolean isComposite(int num) {

if (num <= 1) {

return false;

}

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

if (num % i == 0) {

return true;

}

}

return false;

}

public static int countCompositeNumbers(int[] numbers) {

int count = 0;

for (int num : numbers) {

if (isComposite(num)) {

count++;

}

}

return count;

}

}

30.program to check whether the given character is present in the string

import java.util.Scanner;

public class CharacterCheck {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String inputString = scanner.nextLine();

System.out.print("Enter a character to check: ");

char characterToCheck = scanner.next().charAt(0);

boolean isPresent = checkCharacterInString(inputString, characterToCheck);

if (isPresent) {

System.out.println("The character '" + characterToCheck + "' is present in the string.");

} else {

System.out.println("The character '" + characterToCheck + "' is not present in the string.");

}

scanner.close();

}

public static boolean checkCharacterInString(String str, char ch) {

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

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

return true;

}

}

return false;

}

}

32.armstrong number or not

import java.util.Scanner;

public class ArmstrongNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

if (isArmstrongNumber(number)) {

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

} else {

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

}

scanner.close();

}

public static boolean isArmstrongNumber(int number) {

int originalNumber = number;

int sum = 0;

int numberOfDigits = String.valueOf(number).length();

while (number != 0) {

int digit = number % 10;

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

number /= 10;

}

return sum == originalNumber;

}

}

33.sum of digits of n digit number

import java.util.Scanner;

public class SumOfDigits {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

long number = scanner.nextLong();

long sum = calculateSumOfDigits(number);

System.out.println("The sum of the digits of " + number + " is: " + sum);

scanner.close();

}

public static long calculateSumOfDigits(long number) {

long sum = 0;

while (number != 0) {

sum += number % 10;

number /= 10;

}

return sum;

}

}

34.square root of perfect square number

import java.util.Scanner;

public class PerfectSquareRoot {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

double squareRoot = calculateSquareRoot(number);

System.out.println("The square root of " + number + " is: " + squareRoot);

scanner.close();

}

public static double calculateSquareRoot(int number) {

return Math.sqrt(number);

}

}

35.inverted pyramid pattern

same as 17th

36.count all prime and composite numbers entered by user

import java.util.Scanner;

public class PrimeAndCompositeCounter {

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 primeCount = 0;

int compositeCount = 0;

for (int number : numbers) {

if (isPrime(number)) {

primeCount++;

} else if (isComposite(number)) {

compositeCount++;

}

}

System.out.println("Number of prime numbers: " + primeCount);

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

scanner.close();

}

public static boolean isPrime(int num) {

if (num <= 1) {

return false;

}

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

if (num % i == 0) {

return false;

}

}

return true;

}

public static boolean isComposite(int num) {

if (num <= 1) {

return false;

}

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

if (num % i == 0) {

return true;

}

}

return false;

}

}

37.mth maximum and nth minimum number in an array and fin its sum an difference

import java.util.Arrays;

import java.util.Scanner;

public class MaxMinSumDifference {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int size = scanner.nextInt();

int[] numbers = new int[size];

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

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

numbers[i] = scanner.nextInt();

}

System.out.print("Enter the value of m (for m-th maximum): ");

int m = scanner.nextInt();

System.out.print("Enter the value of n (for n-th minimum): ");

int n = scanner.nextInt();

Arrays.sort(numbers);

int mthMax = numbers[size - m];

int nthMin = numbers[n - 1];

int sum = mthMax + nthMin;

int difference = mthMax - nthMin;

System.out.println("The " + m + "-th maximum number is: " + mthMax);

System.out.println("The " + n + "-th minimum number is: " + nthMin);

System.out.println("The sum of the " + m + "-th maximum and the " + n + "-th minimum is: " + sum);

System.out.println("The difference between the " + m + "-th maximum and the " + n + "-th minimum is: " + difference);

scanner.close();

}

}

38.write a program to print total amount available in ATM machines with conditions applied. Total denominations are 2000,500,200,100 get the denomination priority rom the user and the total number of notes from the user

import java.util.Scanner;

public class ATM {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input priority of denominations

int[] denominations = new int[4];

System.out.println("Enter the priority of denominations (2000, 500, 200, 100): ");

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

denominations[i] = scanner.nextInt();

}

// Input total number of notes for each denomination

int[] notes = new int[4];

System.out.println("Enter the total number of notes for each denomination:");

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

System.out.print("Denomination " + denominations[i] + ": ");

notes[i] = scanner.nextInt();

}

// Calculate total amount

int totalAmount = 0;

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

totalAmount += denominations[i] \* notes[i];

}

// Print total amount

System.out.println("Total amount available in the ATM: " + totalAmount);

scanner.close();

}

}

39.case 1:given string is palindrome or not

case 2: given number is palindrome or not using choice

import java.util.Scanner;

public class PalindromeCheckWithChoice {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Choose an option:");

System.out.println("1. Check if a given string is a palindrome.");

System.out.println("2. Check if a given number is a palindrome.");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

switch (choice) {

case 1:

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

String inputString = scanner.nextLine();

if (isPalindrome(inputString)) {

System.out.println("The string \"" + inputString + "\" is a palindrome.");

} else {

System.out.println("The string \"" + inputString + "\" is not a palindrome.");

}

break;

case 2:

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

int inputNumber = scanner.nextInt();

if (isPalindrome(inputNumber)) {

System.out.println("The number " + inputNumber + " is a palindrome.");

} else {

System.out.println("The number " + inputNumber + " is not a palindrome.");

}

break;

default:

System.out.println("Invalid choice. Please choose 1 or 2.");

}

scanner.close();

}

// Method to check if a given string is a 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;

}

// Method to check if a given number is a palindrome

public static boolean isPalindrome(int number) {

int originalNumber = number;

int reversedNumber = 0;

while (number != 0) {

int digit = number % 10;

reversedNumber = reversedNumber \* 10 + digit;

number /= 10;

}

return originalNumber == reversedNumber;

}

}

40.decimal number to binary number

import java.util.Scanner;

public class DecimalToBinary {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int decimalNumber = scanner.nextInt();

String binaryNumber = convertToBinary(decimalNumber);

System.out.println("The binary representation of " + decimalNumber + " is: " + binaryNumber);

scanner.close();

}

public static String convertToBinary(int decimalNumber) {

if (decimalNumber == 0) {

return "0";

}

StringBuilder binary = new StringBuilder();

while (decimalNumber > 0) {

binary.insert(0, decimalNumber % 2);

decimalNumber /= 2;

}

return binary.toString();

}

}

41: Employee Bonus Calculation

import java.util.Scanner;

public class EmployeeBonus {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input: Grade and Salary

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

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

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

double salary = scanner.nextDouble();

double bonus = 0;

// Calculate bonus based on grade

if (grade == 'A' || grade == 'a') {

bonus = salary \* 0.05;

} else if (grade == 'B' || grade == 'b') {

bonus = salary \* 0.10;

}

// Extra bonus for salaries under $10,000

if (salary < 10000) {

bonus += salary \* 0.02;

}

// Total salary after bonus

double totalSalary = salary + bonus;

// Output results

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

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

System.out.println("Total to be paid = " + totalSalary);

scanner.close();

}

}

42: Read Marks for Four Subjects and Calculate Grade

import java.util.Scanner;

public class StudentGrade {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int subjects = 4;

int total = 0;

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

System.out.print("Enter marks for subject " + i + ": ");

total += scanner.nextInt();

}

double aggregate = total / (double) subjects;

String grade;

if (aggregate > 75) grade = "Distinction";

else if (aggregate >= 60) grade = "First Division";

else if (aggregate >= 50) grade = "Second Division";

else if (aggregate >= 40) grade = "Third Division";

else grade = "Fail";

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

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

System.out.println("Grade = " + grade);

scanner.close();

}

}

43: Find 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 / 2; i++) {

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

}

return sum == num;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

int count = 0, num = 1;

while (count < n) {

if (isPerfect(num)) {

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

count++;

}

num++;

}

scanner.close();

}

}

44: Fibonacci Series and Sum of Even Indexed Elements

import java.util.Scanner;

public class FibonacciEvenSum {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter N (length of Fibonacci series): ");

int n = scanner.nextInt();

int[] fib = new int[n];

fib[0] = 0;

fib[1] = 1;

int sumEvenIndexes = fib[0];

System.out.print("Fibonacci Series: " + fib[0] + ", " + fib[1]);

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

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

System.out.print(", " + fib[i]);

if (i % 2 == 0) sumEvenIndexes += fib[i];

}

System.out.println("\nSum of numbers at even indexes: " + sumEvenIndexes);

scanner.close();

}

}

45: Read Numbers Until -1 Is Encountered

import java.util.Scanner;

public class ReadUntilNegativeOne {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int num;

do {

System.out.print("Enter a number (-1 to stop): ");

num = scanner.nextInt();

} while (num != -1);

System.out.println("End of input.");

scanner.close();

}

}

46: Read Characters Until \* Is Encountered

java

import java.util.Scanner;

public class ReadUntilAsterisk {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

char ch;

do {

System.out.print("Enter a character (\* to stop): ");

ch = scanner.next().charAt(0);

} while (ch != '\*');

System.out.println("End of input.");

scanner.close();

}

}

47: Find Nth Largest Element in an Array

import java.util.Arrays;

import java.util.Scanner;

public class NthLargestElement {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

int[] arr = new int[n];

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

System.out.print("Enter element " + (i + 1) + ": ");

arr[i] = scanner.nextInt();

}

System.out.print("Enter value of N (position of largest element): ");

int nth = scanner.nextInt();

Arrays.sort(arr);

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

scanner.close();

}

}

48: Compute Student Grades Using Multiple Inheritance (Using Interfaces)

import java.util.Scanner;

interface Marks {

void inputMarks();

}

interface Results {

void displayResult();

}

class Student implements Marks, Results {

private int[] marks = new int[6];

private int total;

private double aggregate;

private String grade;

public void inputMarks() {

Scanner scanner = new Scanner(System.in);

String[] subjects = {"Python", "C Programming", "Mathematics", "Physics", "Chemistry", "Professional Ethics"};

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

System.out.print("Enter marks in " + subjects[i] + ": ");

marks[i] = scanner.nextInt();

}

calculate();

}

private void calculate() {

total = 0;

for (int mark : marks) total += mark;

aggregate = total / 6.0;

if (aggregate > 75) grade = "Distinction";

else if (aggregate >= 60) grade = "First Division";

else if (aggregate >= 50) grade = "Second Division";

else if (aggregate >= 40) grade = "Third Division";

else grade = "Fail";

}

public void displayResult() {

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

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

System.out.println("Grade = " + grade);

}

public static void main(String[] args) {

Student student = new Student();

student.inputMarks();

student.displayResult();

}

}

50: Bank Interest Rate using 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: " + sbi.getRateOfInterest());

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

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

}

}

51: Member Name Hiding in Subclass

class Parent {

int num = 10;

}

class Child extends Parent {

int num = 20; // Hides parent variable

void display() {

System.out.println("Child num: " + num);

System.out.println("Parent num: " + super.num);

}

}

public class VariableHiding {

public static void main(String[] args) {

Child obj = new Child();

obj.display();

}

}

52: Fibonacci Series using Runnable Interface

class FibonacciThread implements Runnable {

private int n;

FibonacciThread(int n) {

this.n = n;

}

public void run() {

int a = 0, b = 1, c;

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

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

c = a + b;

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

a = b;

b = c;

}

System.out.println();

}

}

public class FibonacciRunnable {

public static void main(String[] args) {

int n = 10; // Example input

Thread t = new Thread(new FibonacciThread(n));

t.start();

}

}

53: Sum of N Numbers with ArrayIndexOutOfBoundsException Handling

import java.util.Scanner;

public class SumNumbers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

int[] numbers = new int[n];

int sum = 0;

try {

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

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

numbers[i] = scanner.nextInt();

sum += numbers[i];

}

} catch (ArrayIndexOutOfBoundsException e) {

System.out.println("Array Index Out of Bounds!");

}

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

scanner.close();

}

}

54: Prime Number Check using Runnable Interface

class PrimeChecker implements Runnable {

private int num;

PrimeChecker(int num) {

this.num = num;

}

public void run() {

if (num < 2) {

System.out.println(num + " is not Prime.");

return;

}

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

if (num % i == 0) {

System.out.println(num + " is not Prime.");

return;

}

}

System.out.println(num + " is Prime.");

}

}

public class PrimeThread {

public static void main(String[] args) {

int num = 13; // Example input

Thread t = new Thread(new PrimeChecker(num));

t.start();

}

}

55: Convert Days to Years, Weeks, and Days

import java.util.Scanner;

public class ConvertDays {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int days = scanner.nextInt();

if (days < 0) {

System.out.println("Invalid input. Days cannot be negative.");

} else {

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);

}

scanner.close();

}

}

56: Find the Number of Student Users in a College

import java.util.Scanner;

public class CollegeUsers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int totalUsers = scanner.nextInt();

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

int staffUsers = scanner.nextInt();

if (totalUsers < 0 || staffUsers < 0 || staffUsers > totalUsers) {

System.out.println("Invalid Input");

} else {

int nonTeachingStaff = staffUsers / 3;

int studentUsers = totalUsers - (staffUsers + nonTeachingStaff);

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

}

scanner.close();

}

}

57: Find Number of Factors and Nth Factor

import java.util.Scanner;

public class Factors {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int num = scanner.nextInt();

System.out.print("Enter N (Nth Factor to find): ");

int n = scanner.nextInt();

int count = 0;

int nthFactor = -1;

System.out.print("Factors of " + num + ": ");

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

if (num % i == 0) {

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

count++;

if (count == n) nthFactor = i;

}

}

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

if (nthFactor != -1) System.out.println(n + "th factor is: " + nthFactor);

else System.out.println("Invalid N value");

scanner.close();

}

}

58: Print n Prime Numbers After the nth Prime

import java.util.Scanner;

public class NextNPrimes {

public static boolean isPrime(int num) {

if (num < 2) return false;

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

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

}

return true;

}

public static int findNthPrime(int n) {

int count = 0, num = 1;

while (count < n) {

num++;

if (isPrime(num)) count++;

}

return num;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter N (Find nth Prime): ");

int n = scanner.nextInt();

System.out.print("Enter count of primes to print after nth Prime: ");

int count = scanner.nextInt();

int nthPrime = findNthPrime(n);

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

System.out.print(count + " primes after " + nthPrime + " are: ");

int nextPrime = nthPrime + 1, printed = 0;

while (printed < count) {

if (isPrime(nextPrime)) {

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

printed++;

}

nextPrime++;

}

scanner.close();

}

}

59: Find Perfect Squares in a Range with Digit Sum < 10

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 scanner = new Scanner(System.in);

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

int lower = scanner.nextInt();

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

int upper = scanner.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 + " ");

}

}

scanner.close();

}

}

60: Print Unique Permutations of a Given Number

import java.util.Arrays;

import java.util.HashSet;

import java.util.Scanner;

import java.util.Set;

public class Permutations {

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

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

uniquePerms.add(ans);

return;

}

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

char ch = str.charAt(i);

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

generatePermutations(ros, ans + ch, uniquePerms);

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String num = scanner.next();

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

generatePermutations(num, "", uniquePerms);

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

for (String perm : uniquePerms) {

System.out.println(perm);

}

scanner.close();

}

}

61: Create an Array with Each Number and Its Square

import java.util.Scanner;

public class NumberSquareArray {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int lower = scanner.nextInt();

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

int upper = scanner.nextInt();

if (lower > upper) {

System.out.println("Invalid range.");

} else {

System.out.print("[");

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

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

if (i < upper) System.out.print(", ");

}

System.out.println("]");

}

scanner.close();

}

}

62: Bank Account System

import java.util.Scanner;

class BankAccount {

String depositorName;

int accountNumber;

String accountType;

double balance = 10000; // Default balance

void createAccount() {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Depositor Name: ");

depositorName = scanner.nextLine();

System.out.print("Enter Account Number: ");

accountNumber = scanner.nextInt();

System.out.print("Enter Account Type (Savings/Current): ");

accountType = scanner.next();

}

void deposit(double amount) {

balance += amount;

System.out.println("Amount deposited successfully. New balance: " + balance);

}

void withdraw(double amount) {

if (balance - amount >= 500) {

balance -= amount;

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

} else {

System.out.println("Insufficient balance. Minimum balance of Rs.500 must be maintained.");

}

}

void displayBalance() {

System.out.println("Account Holder: " + depositorName);

System.out.println("Account Number: " + accountNumber);

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

}

}

public class BankSystem {

public static void main(String[] args) {

BankAccount account = new BankAccount();

account.createAccount();

account.deposit(5000);

account.withdraw(3000);

account.displayBalance();

}

}

63: Reverse and Add Until a Palindrome

import java.util.Scanner;

public class ReverseAndAdd {

public static int reverse(int num) {

int rev = 0;

while (num > 0) {

rev = rev \* 10 + num % 10;

num /= 10;

}

return rev;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int num = scanner.nextInt();

while (true) {

int reversed = reverse(num);

int sum = num + reversed;

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

if (sum == reverse(sum)) {

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

break;

}

num = sum;

}

scanner.close();

}

}

64: Bank Account with Synchronization (Deposit & Withdraw)

class BankAccount {

private int balance = 10000;

public synchronized void withdraw(int amount) {

while (balance < amount) {

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

try {

wait();

} catch (InterruptedException e) {

System.out.println("Interrupted");

}

}

balance -= amount;

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

}

public synchronized void deposit(int amount) {

balance += amount;

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

notify();

}

}

public class BankSynchronization {

public static void main(String[] args) {

BankAccount account = new BankAccount();

Thread withdrawThread = new Thread(() -> account.withdraw(12000));

Thread depositThread = new Thread(() -> account.deposit(3000));

withdrawThread.start();

depositThread.start();

}

}

65: FizzBuzz Implementation

import java.util.Scanner;

public class FizzBuzz {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.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);

}

scanner.close();

}

}

66: Find Common Elements in Two Arrays

import java.util.HashSet;

import java.util.Scanner;

import java.util.Set;

public class CommonElements {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n1 = scanner.nextInt();

int[] arr1 = new int[n1];

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

for (int i = 0; i < n1; i++) {

arr1[i] = scanner.nextInt();

}

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

int n2 = scanner.nextInt();

int[] arr2 = new int[n2];

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

for (int i = 0; i < n2; i++) {

arr2[i] = scanner.nextInt();

}

Set<Integer> set1 = new HashSet<>();

for (int num : arr1) {

set1.add(num);

}

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

for (int num : arr2) {

if (set1.contains(num)) {

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

set1.remove(num);

}

}

System.out.println();

scanner.close();

}

}

67: Convert Roman Numeral to Integer

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class RomanToInteger {

public static int romanToInt(String s) {

Map<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 result = 0;

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

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

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

} else {

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

}

}

return result;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String roman = scanner.next().toUpperCase();

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

scanner.close();

}

}

68: Check if Ransom Note Can Be Formed from Magazine

import java.util.HashMap;

import java.util.Scanner;

public class RansomNote {

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

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

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

map.put(c, map.getOrDefault(c, 0) + 1);

}

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

if (!map.containsKey(c) || map.get(c) == 0) {

return false;

}

map.put(c, map.get(c) - 1);

}

return true;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String ransomNote = scanner.next();

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

String magazine = scanner.next();

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

scanner.close();

}

}

69: Steps to Reduce a Number to Zero

import java.util.Scanner;

public class ReduceToZero {

public static int numberOfSteps(int num) {

int steps = 0;

while (num > 0) {

if (num % 2 == 0) {

num /= 2;

} else {

num -= 1;

}

steps++;

}

return steps;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int num = scanner.nextInt();

System.out.println("Number of steps to reduce to zero: " + numberOfSteps(num));

scanner.close();

}

}

70: Compute Student Grades Using Multiple Inheritance (Using Interfaces)

import java.util.Scanner;

interface Marks {

void inputMarks();

}

interface Results {

void displayResult();

}

class Student implements Marks, Results {

private int[] marks = new int[6];

private int total;

private double aggregate;

private String grade;

public void inputMarks() {

Scanner scanner = new Scanner(System.in);

String[] subjects = {"Python", "C Programming", "Mathematics", "Physics", "Chemistry", "Professional Ethics"};

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

System.out.print("Enter marks in " + subjects[i] + ": ");

marks[i] = scanner.nextInt();

}

calculate();

}

private void calculate() {

total = 0;

for (int mark : marks) total += mark;

aggregate = total / 6.0;

if (aggregate > 75) grade = "Distinction";

else if (aggregate >= 60) grade = "First Division";

else if (aggregate >= 50) grade = "Second Division";

else if (aggregate >= 40) grade = "Third Division";

else grade = "Fail";

}

public void displayResult() {

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

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

System.out.println("Grade = " + grade);

}

public static void main(String[] args) {

Student student = new Student();

student.inputMarks();

student.displayResult();

}

}