1.Write a program to count all the prime and composite numbers entered by the 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 primes = 0, composites = 0;

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

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

int num = scanner.nextInt();

if (isPrime(num))

primes++;

else

composites++;

}

System.out.println("Total prime numbers: " + primes);

System.out.println("Total composite numbers: " + composites);

}

static boolean isPrime(int n) {

if (n <= 1)

return false;

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

if (n % i == 0)

return false;

}

return true;

}

}

Output:

Enter the number of elements: 5

Enter number 1: 3

Enter number 2: 4

Enter number 3: 5

Enter number 4: 6

Enter number 5: 7

Total prime numbers: 3

Total composite numbers: 2

2.Find the Mth maximum number and Nth minimum number in an array and then find the sum of it and difference of it.

import java.util.Arrays;

public class MaxMinSumDiff {

public static void main(String[] args) {

int[] arr = {5, 3, 8, 2, 10, 7};

int m = 2; // Mth maximum

int n = 2; // Nth minimum

Arrays.sort(arr);

int maxM = arr[arr.length - m];

int minN = arr[n - 1];

int sum = maxM + minN;

int diff = maxM - minN;

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

System.out.println("Difference: " + diff);

}

}

Output:

Sum: 11

Difference: 5

1. Write a program to print the total amount available in the ATM machine with the conditions applied.

import java.util.Scanner;

public class ATM {

// Function to calculate total amount in the ATM

public static int calculateTotalAmount(int[] denominations, int[] counts) {

int totalAmount = 0;

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

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

}

return totalAmount;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Array of available denominations

int[] denominations = {1000, 500, 200, 100, 50, 20, 10, 5, 2, 1};

// Array to store the count of each denomination

int[] counts = new int[denominations.length];

// Taking input for counts of each denomination

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

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

counts[i] = scanner.nextInt();

}

// Calculating the total amount

int totalAmount = calculateTotalAmount(denominations, counts);

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

}

}

Output:

Enter the count of 1000 denomination: 10

Enter the count of 500 denomination: 5

Enter the count of 200 denomination: 10

Enter the count of 100 denomination: 20

Enter the count of 50 denomination: 30

Enter the count of 20 denomination: 50

Enter the count of 10 denomination: 40

Enter the count of 5 denomination: 60

Enter the count of 2 denomination: 100

Enter the count of 1 denomination: 200

Total amount available in the ATM machine: 49250

5.Write a program using choice to check

Case 1: Given string is palindrome or not

Case 2: Given number is palindrome or not

import java.util.Scanner;

public class PalindromeChecker {

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 string is palindrome");

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

int choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1:

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

String str = scanner.nextLine();

if (isPalindrome(str))

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

else

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

break;

case 2:

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

int num = scanner.nextInt();

if (isPalindrome(num))

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

else

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

break;

default:

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

}

}

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;

}

return true;

}

static boolean isPalindrome(int num) {

int original = num;

int reversed = 0;

while (num != 0) {

int digit = num % 10;

reversed = reversed \* 10 + digit;

num /= 10;

}

return original == reversed;

}

}

Output:

Choose an option:

1. Check if a string is palindrome

2. Check if a number is palindrome

1

Enter a string: malayalam

The string is a palindrome.

5.Write a program to convert Decimal number equivalent to Binary number and octal       numbers?

import java.util.Scanner;

public class DecimalToBinaryOctal {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int decimal = scanner.nextInt();

String binary = Integer.toBinaryString(decimal);

String octal = Integer.toOctalString(decimal);

System.out.println("Binary equivalent: " + binary);

System.out.println("Octal equivalent: " + octal);

}

}

Output:

Enter a decimal number: 9

Binary equivalent: 1001

Octal equivalent: 11