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);

int countPrime = 0, countComposite = 0;

while (true) {

System.out.print("Enter a number or type -1 to exit: ");

int num = scanner.nextInt();

if (num == -1) {

break;

}

if (isPrime(num)) {

countPrime++;

} else {

countComposite++;

}

}

System.out.printf("Total prime numbers: %d%n", countPrime);

System.out.printf("Total composite numbers: %d%n", countComposite);

}

private static boolean isPrime(int num) {

if (num <= 1) {

return false;

}

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

if (num % i == 0) {

return false;

}

}

return true;

}

}

Output:

Enter a number or type -1 to exit: 5

Enter a number or type -1 to exit: 7

Enter a number or type -1 to exit: 9

Enter a number or type -1 to exit: 11

Enter a number or type -1 to exit: 12

Enter a number or type -1 to exit: 13

Enter a number or type -1 to exit: -1

Total prime numbers: 4

Total composite numbers: 1

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;

import java.util.Scanner;

public class MthMaxNthMinSumDifference {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

int[] arr = new int[n];

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

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

arr[i] = scanner.nextInt();

}

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

int m = scanner.nextInt();

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

int nth = scanner.nextInt();

if (m > n || nth > n) {

System.out.println("M and/or N are greater than the array size.");

return;

}

Arrays.sort(arr);

int mthMax = arr[n - m];

int nthMin = arr[nth - 1];

int sum = mthMax + nthMin;

int difference = mthMax - nthMin;

System.out.printf("Mth maximum number: %d%n", mthMax);

System.out.printf("Nth minimum number: %d%n", nthMin);

System.out.printf("Sum of Mth maximum and Nth minimum: %d%n", sum);

System.out.printf("Difference between Mth maximum and Nth minimum: %d%n", difference);

}

}

Output:

Enter the size of the array: 7

Enter the elements of the array:

1 5 8 2 6 9 3

Enter M: 3

Enter N: 5

Mth maximum number: 8

Nth minimum number: 3

Sum of Mth maximum and Nth minimum: 11

Difference between Mth maximum and Nth minimum: 5

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

import java.util.Scanner;

public class ATMMachine {

public static void main(String[] args) {

double balance = 10000;

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.printf("Current balance: $%.2f%n", balance);

System.out.print("Enter the amount to withdraw ($20, $40, $60, $100, $200) or type -1 to exit: ");

int amount = scanner.nextInt();

if (amount == -1) {

break;

}

if (amount % 20 != 0) {

System.out.println("Invalid amount. Please enter a multiple of 20.");

continue;

}

if (amount > balance) {

System.out.println("Insufficient balance.");

continue;

}

balance -= amount;

System.out.printf("Withdrew $%.2f%n", amount);

System.out.printf("Remaining balance: $%.2f%n", balance);

}

System.out.println("Thank you for using the ATM.");

}

}

Output:

Current balance: $10000.00

Enter the amount to withdraw ($20, $40, $60, $100, $200) or type -1 to exit: 200

Withdrew $200.00

Remaining balance: $9800.00

Current balance: $9800.00

Enter the amount to withdraw ($20, $40, $60, $100, $200) or type -1 to exit: 60

Withdrew $60.00

Remaining balance: $9740.00

Current balance: $9740.00

Enter the amount to withdraw ($20, $40, $60, $100, $200) or type -1 to exit: 150

Invalid amount. Please enter a multiple of 20.

Current balance: $9740.00

Enter the amount to withdraw ($20, $40, $60, $100, $200) or type -1 to exit: -1

Thank you for using the ATM.

4. 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.print("Enter 1 to check a string or 2 to check a number: ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

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

String str = scanner.next();

if (isPalindrome(str)) {

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

} else {

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

} else {

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

}

break;

default:

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

}

}

private 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;

}

private static boolean isPalindrome(int num) {

int originalNum = num;

int reverseNum = 0;

while (num != 0) {

int remainder = num % 10;

reverseNum = reverseNum \* 10 + remainder;

num/= 10;

}

return originalNum == reverseNum;

}

}

Output:

Enter 1 to check a string or 2 to check a number: 1

Enter a string: racecar

The given string is a palindrome.

Enter 1 to check a string or 2 to check a number: 2

Enter a number: 12321

The given number 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.printf("Binary representation: %s%n", binary);

System.out.printf("Octal representation: %s%n", octal);

}

}

Output:

Enter a decimal number: 10

Binary representation: 1010

Octal representation: 12