1. Write a program to count all the prime and composite numbers entered by the user.

Sample Input:

Enter the numbers

4

54

29

71

7

59

98

23

Sample Output:

Composite number:3

Prime number:5

Test cases:

1. 33, 41, 52, 61,73,90

2. TEN, FIFTY, SIXTY-ONE, SEVENTY-SEVEN, NINE

3. 45, 87, 09, 5.0 ,2.3, 0.4

4. -54, -76, -97, -23, -33, -98

5. 45, 73, 00, 50, 67, 44

```
import java.util.*;


public class CountPrimeComposite {
   public static void main(String args[]) {
      int p = 0, c = 0;


      Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter the number of numbers:");
        int n = sc.nextInt();

        System.out.println("Enter the numbers:");
        for (int i = 0; i < n; i++) {
            int num = sc.nextInt();

            for (int j = 2; j < num; j++) {
                if (num % j == 0) {
                    c += 1;
                    break;
                } else {
                    p += 1;
                    break;
                }
            }
        }

        System.out.println("Count of prime numbers: " + p);
        System.out.println("Count of composite numbers: " + c);
    }
}
package medium;

import java.util.*;
```

```
public class m1 {

    public static void main(String args[]) {

        int p = 0, c = 0;


        Scanner sc = new Scanner(System.in);


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

        int n = sc.nextInt();


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


        try {

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

                int num = sc.nextInt();


                if (num >= 0) {

                    if (num == 0 || num == 1) {

                        c += 1;

                    } else {

                        boolean isComposite = false;


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

                            if (num % j == 0) {

                                isComposite = true;

                                break;

                            }

                        }
```

```
                    if (isComposite) {
                        c += 1;
                    } else {
                        p += 1;
                    }
                }
            } else {
                System.out.println("Enter only positive input");
            }
        } catch (InputMismatchException e) {
            System.out.println("Enter only integers");
        }

        System.out.println("Count of composite numbers: " + c);
        System.out.println("Count of prime numbers: " + p);
    }
}
```

1. Find the M^{th} maximum number and N^{th} minimum number in an array and then find the sum of it and difference of it.

   Sample Input:

   Array of elements = {14, 16, 87, 36, 25, 89, 34}

M = 1

N = 3

Sample Output:

1st Maximum Number = 89

3rd Minimum Number = 25

Sum = 114

Difference = 64

Test cases:

1. {16, 16, 16 16, 16}, M = 0, N = 1

2. {0, 0, 0, 0}, M = 1, N = 2

3. {-12, -78, -35, -42, -85}, M = 3 , N = 3

4. {15, 19, 34, 56, 12}, M = 6 , N = 3

5. {85, 45, 65, 75, 95}, M = 5 , N = 7

```
package medium;

import java.util.Arrays;
import java.util.Scanner;

public class m2 {
   public static void main(String[] args)
   {
      Scanner sc = new Scanner(System.in);

      System.out.println("Enter the number of elements in the array:");
      int length = sc.nextInt();
      int[] arr = new int[length];

      System.out.println("Enter the elements of the array:");
      for (int i = 0; i < length; i++)
      {
         arr[i] = sc.nextInt();
      }

      System.out.println("Enter the value of M:");
      int m = sc.nextInt();

      System.out.println("Enter the value of N:");
      int n = sc.nextInt();

      if (m <= 0 || m > length || n <= 0 || n > length)
      {
         System.out.println("Invalid values for M and N. They should be between 1 and the length of the array.");
      }
      else
      {
          Arrays.sort(arr);
          int mthMax = arr[length - m];
```

```
        int nthMin = arr[n - 1];

        int sum = mthMax + nthMin;
        int diff = mthMax - nthMin;

        System.out.println("Mth Maximum Number = " + mthMax);
        System.out.println("Nth Minimum Number = " + nthMin);
        System.out.println("Sum = " + sum);
        System.out.println("Difference = " + diff);
    }


  }
}
```

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

Total denominations are 2000, 500, 200, 100, get the denomination priority from the user and the total number of notes from the user to display the total available balance to the user

Sample Input:

Enter the 1st Denomination: 500

Enter the 1st Denomination number of notes: 4

Enter the 2nd Denomination: 100

Enter the 2nd Denomination number of notes: 20

Enter the 3rd Denomination: 200

Enter the 3rd Denomination number of notes: 32

Enter the 4th Denomination: 2000

Enter the 4th Denomination number of notes: 1

Sample Output:

Total Available Balance in ATM: 12400

Test Cases:

3 Hidden Test cases (Think Accordingly based on Denominations)

```
package medium;
```

```
import java.util.Scanner;


public class m3 {
    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 < denominations.length; i++)
        {
            System.out.println("Enter the " + (i + 1) + " Denomination:");
            int denomination = sc.nextInt();


            boolean validDenomination = false;
            for (int d : denominations)
            {
                if (d == denomination)
                {
                    validDenomination = true;
                    break;
                }
            }


            if (validDenomination)
            {
                System.out.println("Enter the " + (i + 1) + " Denomination number of notes:");
```

```
            int numberOfNotes = sc.nextInt();


            totalBalance += denomination * numberOfNotes;
        } else
        {
            System.out.println("Invalid denomination. Please enter a valid denomination.");
            i--;
        }
    }

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

1. Write a program using choice to check

    Case 1: Given string is palindrome or not

    Case 2: Given number is palindrome or not

    Sample Input:

    Case = 1

    String = MADAM

    Sample Output:

    Palindrome

    Test cases:

    1. MONEY

    2. 5678765

    3. MALAY12321ALAM

    4. MALAYALAM

    5. 1234.4321

```
package medium;

import java.util.Scanner;

public class m4 {
    public static void main(String[] args) {
        Scanner sc = 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 = sc.nextInt();

        switch (choice) {
            case 1:
                System.out.println("Enter a string:");
                String str = sc.next();
                String reversedStr = new StringBuilder(str).reverse().toString();
                if (str.equalsIgnoreCase(reversedStr)) {
                    System.out.println("Palindrome");
                } else {
                    System.out.println("Not a Palindrome");
                }
                break;

            case 2:
```

```
            System.out.println("Enter a number:");
            int num = sc.nextInt();
            int originalNum = num;
            int reversedNum = 0;
            while (num != 0) {
                int digit = num % 10;
                reversedNum = reversedNum * 10 + digit;
                num /= 10;
            }
            if (originalNum == reversedNum) {
                System.out.println("Palindrome");
            } else {
                System.out.println("Not a Palindrome");
            }
            break;

        default:
            System.out.println("Invalid choice");
    }

  }
}
```

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

Sample Input:

Decimal Number: 15

Sample Output:

Binary Number = 1111

Octal = 17

Test cases:

1. 111

2. 15.2

3. 0

4. B12

5. 1A.2

```
package medium;


import java.util.Scanner;


public class m5 {
   public static void main(String[] args)
   {
      Scanner sc = new Scanner(System.in);


      System.out.println("Enter a decimal number:");
      if (sc.hasNextInt())
      {
         int decimalNumber = sc.nextInt();



         String binaryNumber = Integer.toBinaryString(decimalNumber);
         System.out.println("Binary Number = " + binaryNumber);
```

```
            String octalNumber = Integer.toOctalString(decimalNumber);

            System.out.println("Octal = " + octalNumber);

        } else

        {

            System.out.println("Invalid input. Please enter a valid decimal number.");

        }



    }

}
```

6. In an organization they decide to give bonus to all the employees on New Year. A 5% bonus on salary is given to the grade A workers and 10% bonus on salary to the grade B workers. Write a program to enter the salary and grade of the employee. If the salary of the employee is less than $10,000 then the employee gets an extra 2% bonus on salary Calculate the bonus that has to be given to the employee and print the salary that the employee will get.

Sample Input & Output:

Enter the grade of the employee: B

Enter the employee salary: 50000

Salary=50000

Bonus=5000.0

Total to be paid:55000.0

Test cases:

1. Enter the grade of the employee: A

Enter the employee salary: 8000

2. Enter the grade of the employee: C

Enter the employee salary: 60000

3. Enter the grade of the employee: B

Enter the employee salary: 0

4. Enter the grade of the employee: 38000

Enter the employee salary: A

5. Enter the grade of the employee: B

Enter the employee salary: -8000

1. Write a program to print the first n perfect numbers. (Hint Perfect number means **a positive integer that is equal to the sum of its proper divisors)**

Sample Input:

N = 3

Sample Output:

First 3 perfect numbers are: 6 , 28 , 496

Test Cases:

1. N = 0

2. N = 5

3. N = -2

4. N = -5

5. N = 0.2

```
package medium;


import java.util.Scanner;


public class m7 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);


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

        if (sc.hasNextInt()) {

            int n = sc.nextInt();
```

```
    if (n >= 0) {
        int count = 0;
        int num = 2; // Start checking from 2 as 1 is not a perfect number

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

        while (count < n) {
            int sum = 1;
            for (int i = 2; i * i <= num; i++) {
                if (num % i == 0) {
                    sum += i;
                    if (i != num / i) {
                        sum += num / i;
                    }
                }
            }

            if (sum == num) {
                System.out.print(num + " , ");
                count++;
            }
            num++;
        }
    } else {
        System.out.println("Please enter a non-negative integer for N.");
    }
} else {
```

```
            System.out.println("Invalid input. Please enter a valid integer for N.");
        }


        sc.close();
    }
}
```

8. Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is 60>= and <75, then the grade is First Division. If aggregate is 50 >= and <60, then the grade is Second Division. If aggregate is 40>= and <50, then the grade is Third Division. Else the grade is Fail.

Sample Input & Output:

Enter the marks in python: 90

Enter the marks in c programming: 91

Enter the marks in Mathematics: 92

Enter the marks in Physics: 93

Total= 366

Aggregate = 91.5

DISTINCTION

Test cases:

a) 18, 76,93,65

b) 73,78,79,75

c) 98,106,120,95

d) 96,73, -85,95

e) 78,59.8,76,79

```
package medium;

import java.util.Scanner;

public class m8 {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      System.out.println("Enter the marks in Python:");
      double pythonMarks = sc.nextDouble();

      System.out.println("Enter the marks in C Programming:");
      double cProgrammingMarks = sc.nextDouble();

      System.out.println("Enter the marks in Mathematics:");
      double mathematicsMarks = sc.nextDouble();

      System.out.println("Enter the marks in Physics:");
      double physicsMarks = sc.nextDouble();

      // Validate input marks
      if (pythonMarks >= 0 && pythonMarks <= 100 && cProgrammingMarks >= 0 && cProgrammingMarks <= 100
            && mathematicsMarks >= 0 && mathematicsMarks <= 100 && physicsMarks >= 0 && physicsMarks <=
100) {

         double totalMarks = pythonMarks + cProgrammingMarks + mathematicsMarks + physicsMarks;
         double aggregate = totalMarks / 4.0;

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

         if (aggregate > 75) {
            System.out.println("DISTINCTION");
         } else if (aggregate >= 60 && aggregate < 75) {
            System.out.println("First Division");
         } else if (aggregate >= 50 && aggregate < 60) {
            System.out.println("Second Division");
         } else if (aggregate >= 40 && aggregate < 50) {
            System.out.println("Third Division");
         } else {
            System.out.println("Fail");
         }
      } else
      {
         System.out.println("Invalid marks. Please enter valid marks between 0 and 100.");
      }

      sc.close();
   }
}
```

1. Write a program to read the numbers until -1 is encountered. Find the average of positive numbers and negative numbers entered by user.

Sample Input:

Enter -1 to exit…

Enter the number: 7

Enter the number: -2

Enter the number: 9

Enter the number: -8

Enter the number: -6

Enter the number: -4

Enter the number: 10

Enter the number: -1

Sample Output:

The average of negative numbers is: -5.0

The average of positive numbers is : 8.66666667

Test cases:

1. -1,43, -87, -29, 1, -9

2. 73, 7-6,2,10,28,-1

3. -5, -9, -46,2,5,0

4. 9, 11, -5, 6, 0,-1

5. -1,-1,-1,-1,-1

```
package medium;

import java.util.Scanner;
```

```
public class m9 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter -1 to exit...");

        int num;
        int positiveCount = 0;
        int negativeCount = 0;
        int positiveSum = 0;
        int negativeSum = 0;

        while (true) {
            System.out.println("Enter the number:");
            num = sc.nextInt();

            if (num == -1) {
                break;
            }

            if (num >= 0) {
                positiveSum += num;
                positiveCount++;
            } else {
                negativeSum += num;
                negativeCount++;
            }
        }

        if (positiveCount > 0) {
            double positiveAverage = (double) positiveSum / positiveCount;
            System.out.println("The average of positive numbers is: " + positiveAverage);
        } else {
            System.out.println("No positive numbers entered.");
        }

        if (negativeCount > 0) {
            double negativeAverage = (double) negativeSum / negativeCount;
            System.out.println("The average of negative numbers is: " + negativeAverage);
        } else {
            System.out.println("No negative numbers entered.");
        }

        sc.close();
    }
}
```

1. Write a program to read a character until a ***** is encountered. Also count the number of uppercase, lowercase, and numbers entered by the users.

Sample Input:

Enter * to exit…

Enter any character: W

Enter any character: d

Enter any character: A

Enter any character: G

Enter any character: g

Enter any character: H

Enter any character: *

Sample Output:

Total count of lower case:2

Total count of upper case:4

Total count of numbers =0

Test cases:

1. 1,7,6,9,5

2. S, Q, I, K,7, j, M

3. M, j, L, &, @, G

4. D, K, I, 6, L, *

5. *, K, A, e, 1, 8, %, *