1. Write a Java Program to Convert a Given Number of Days in Terms of Years, Weeks & Days.

    Sample Input&Output::

    Enter the number of days: 756

    No. of years: 2

    No. of weeks: 3

    No. of days: 5

    Test cases:

    1. 38
    2. 3.6
    3. 0
    4. -365
    5. -45

    Given a date, return the corresponding day of the week for that date.

    The input is given as three integers representing the day, month and year respectively.

    Return the answer as one of the following values {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}.

    Example 1:

    Input: day = 31, month = 8, year = 2019

    Output: "Saturday"

    Example 2:

    Input: day = 18, month = 7, year = 1999

    Output: "Sunday"

    Example 3:

    Input: day = 15, month = 8, year = 1993

    Output: "Sunday"

    Constraints:

- The given dates are valid dates between the years 1971 and 2100.

2. Write a program to find the number of student users in the college, get the total users, staff users details from the client. Note for every 3 staff user there is one Non teaching staff user assigned by default.

    Sample Input:

    Total Users: 856

    Staff Users: 126

    Sample Output:

    Student Users: 688

    Test Cases:

    1. Total User: 0
    2. Total User: -143
    3. Total User: 1026, Staff User: 1026
    4. Total User: 450, Staff User: 540
    5. Total User: 600, Staff User: 450

3. Write a program to print number of factors and to print nth factor of the given number.

Sample Input:

Given Number: 100

N = 4

Sample Output:

Number of factors = 9

4th factor of 100 = 5

Test Cases:

1. Given Number = 512 , N = 6
2. Given Number = 343 , N = 7
3. Given Number = 1024 , N = 0
4. Given Number = -6561 , N = 3
5. Given Number = 0 , N = 2

4. Write a program to print n prime numbers after n^{th} Prime number

Sample Input:

N = 3

Sample Output:

3rd Prime number is 5

3 prime numbers after 5 are: 7, 11, 13

Test cases:

1. N = P
2. N = 0
3. N = -4
4. N = 11
5. N = 7.2

5. Write a Program to create a list of all numbers in a range which are perfect squares and the sum of the digits of the number is less than 10.

Sample Input & Output:

Enter lower range: 1

Enter upper range: 40

[1, 4, 9, 16, 25, 36]

Test case:

1. Enter lower range: 50

Enter upper range: 100

2. Enter lower range: 5

Enter upper range: 8

3. Enter lower range: 10

Enter upper range: 5

4. Enter lower range: 500

Enter upper range: 500

5. Enter lower range: 0

Enter upper range: -100

6. Write a program to print unique permutations of a given number

Sample Input:

    Given Number: 143

Sample Output:

    Permutations are:

134

143

314

341

413

431

Test cases:

    1. 0

    2. 111

    3. 505

    4. -143

    5. -598

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

    Sample Input:

        Enter the lower range:45

        Enter the upper range:49

    Sample Output:

        [(45, 2025), (46, 2116), (47, 2209), (48, 2304), (49, 2401)]

    Test case:

1. Enter lower range: 50

        Enter upper range: 100

2. Enter lower range: 5

        Enter upper range: 8

3. Enter lower range: 10

        Enter upper range: 5

4. Enter lower range: 500

        Enter upper range: 500

5. Enter lower range: 0

        Enter upper range: -100

8. Develop a JAVA code to display the balance. Include the following members:

- Design a class to represent a bank account.
- **Data Members:** Name of the depositor, Account number, Type of account(Savings/Current), Balance amount in the account(Minimum balance is Rs. 500. 00)
- **Methods:**
    1. To read account number, Depositor name, Type of account.
    2. To deposit an amount (Deposited amount should be added with it)
    3. To withdraw an amount after checking balance(Minimum balance must be Rs. 500. 00

    Note : Assume that balance amount = 10000

Test Cases

1. 100, Raja, S, 8000
2. Raja, 100, S, 9000
3. 101, Rani, S, 12000
4. 102, Ragu, W, 8000
5. 103, Ravi, C, 10000

9. Develop a code to Reverse and Add a Number until you get a Palindrome.
   Sample Input If 7325 is input number, then
   7325 (Input Number) + 5237 (Reverse Of Input Number) = 12562
   12562 + 26521 = 39083
   39083 + 38093 = 77176
   77176 + 67177 = 144353
   144353 + 353441 = 497794 (Palindrome)

Test Cases
   1. 8765
   2. -8765
   3. 0
   4. EIGHT FIVE
   5. 87.57

10. Create Customer class with deposit() and withdraw() as synchronized methods. Declare AccountNo, AccName and Balance as Instance Variables inside the class. From the main class, Input the amount for withdraw() operation and if requested amount is not available in existing Balance amount, withdraw() method should be temporarily suspended using wait() method until deposit() method receives the input for amount, to be added in the existing Balance amount and then withdraw() would be completed in a successful manner. Develop the above scenario using Synchronization and Inter-Thread Communication.
    Note : existing Bank balance amount 10000
    Sample Input : 12000, 3000
    Sample Output : Withdraw operation success, balance amount 1000

Test Cases
   1. 11000, 4000
   2. -10000, -2000
   3. 0, 0
   4. EIGHT SEVEN, FIVE
   5. 100.67, 200.68

11. Given an integer n, return a string array answer (1-indexed) where:
    answer(i) == "FizzBuzz" if i is divisible by 3 and 5.
    answer(i) == "Fizz" if i is divisible by 3.
    answer(i) == "Buzz" if i is divisible by 5.
    answer(i) == i (as a string) if none of the above conditions are **true.**

    Example 1:
    Input: n = 3
    Output: ["1", "2", "Fizz"]

Test Case

Test Case	Inputs
1.	n = 5
2.	n = 10
3.	n = 12
4.	n = 18
5.	n = 20

12. Write a Java program to find the common elements in two array of Positive integer

Sample Input：

[1, 2, 3, 4]

[2, 4, 5, 6, 7]

Expected output：[2, 4]

Test Case

Test Case	Inputs-1	Inputs-2
1.	[1, 2, 3, 4]	[4, 5, 6, 7, 8]
2.	[a, b, c, d]	[a, b, c, d]
3.	[1, -2, 3, 4]	[1, -2, 5, 7, 8]
4.	[@, #, 34, 45]	[@, #, %, $,]
5.	[45, 78, 56, 89]	[92, 34, 56, -78, -90]

13. Given a string s consisting of words and spaces, return *the length of the* **last** *word in the string.* A **word** is a maximal substring consisting of non-space characters only. There will be at least one word, consists of only English letters and spaces ' '.

Example 1：

Input： s = "Hello World"

Output： 5

Explanation： The last word is "World" with length 5.

Test Case

Test Case	Inputs-1
1.	Maximal Substring Consisting
2.	**lea@st one wor2d**
3.	1254 98076
4.	& * ( ) % # $
5.	letters and spaces

14. Roman numerals are represented by seven different symbols： I, V, X, L, C, D and M.

SymbolValue

I   1

V   5

X   10

L   50

C   100

| D | 500 |
| M | 1000 |

For example, 2 is written as II in Roman numeral, just two ones added together. 12 is written as XII, which is simply X + II. The number 27 is written as XXVII, which is XX + V + II.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

- I can be placed before V (5) and X (10) to make 4 and 9.
- X can be placed before L (50) and C (100) to make 40 and 90.
- C can be placed before D (500) and M (1000) to make 400 and 900. Given a roman numeral, convert it to an integer.

Example:
Input: s = "III"
Output: 3

Test Case	Inputs
1.	LVIII
2.	MCMXCI
3.	V
4.	LZAII
5.	MCCDTIV

15. Given two strings ransomNote and magazine, return true if ransomNote can be constructed by using the letters from magazine and false otherwise. Each letter in magazine can only be used once in ransomNote.

Example 1:
Input: ransomNote = "a", magazine = "b"
Output: false
Test Case

Test Case	Inputs
1.	ransomNote = "aa", magazine = "ab"
2.	ransomNote = "aa", magazine = "aab"
3.	ransomNote = "abc", magazine = "abc"
4.	ransomNote = "good", magazine = "better"
5.	ransomNote = "xyz", magazine = "123"

16. You are given an m x n binary matrix mat of 1's (representing soldiers) and 0's (representing civilians). The soldiers are positioned in front of the civilians. That is, all the 1's will appear to the left of all the 0's in each row.

A row i is weaker than a row j if one of the following is true:

The number of soldiers in row i is less than the number of soldiers in row j.

Both rows have the same number of soldiers and i< j. Return the indices of the k weakest rows in the matrix ordered from weakest to strongest.

Example 1:
Input: mat =
[[1, 1, 0, 0, 0],
 [1, 1, 1, 1, 0],
 [1, 0, 0, 0, 0],
 [1, 1, 0, 0, 0],
 [1, 1, 1, 1, 1]],
k = 3
Output: [2, 0, 3]
Explanation:
The number of soldiers in each row is:
- Row 0: 2
- Row 1: 4
- Row 2: 1
- Row 3: 2
- Row 4: 5
The rows ordered from weakest to strongest are [2, 0, 3, 1, 4].

Example 2:
Input: mat =
[[1, 0, 0, 0],
 [1, 1, 1, 1],
 [1, 0, 0, 0],
 [1, 0, 0, 0]],
        k = 2
Output: [0, 2]
Explanation:
The number of soldiers in each row is:
- Row 0: 1
- Row 1: 4
- Row 2: 1
- Row 3: 1
The rows ordered from weakest to strongest are [0, 2, 3, 1].

17. Given an integer num, return the number of steps to reduce it to zero. In one step, if the current number is even, you have to divide it by 2, otherwise, you have to subtract 1 from it.

Example 1:
Input: num = 14
Output: 6
Explanation:
Step 1) 14 is even; divide by 2 and obtain 7.
Step 2) 7 is odd; subtract 1 and obtain 6.
Step 3) 6 is even; divide by 2 and obtain 3.
Step 4) 3 is odd; subtract 1 and obtain 2.
Step 5) 2 is even; divide by 2 and obtain 1.

Step 6) 1 is odd; subtract 1 and obtain 0.

Test Case

Test Case	Inputs
1.	n = 5
2.	n = 10
3.	n = 12
4.	n = 18
5.	n = 20

18. Develop a programme that uses Multiple Inheritance concepts to compute a student's grades in six subjects. The total and aggregate are then calculated, and the student's grade is displayed. If the student achieves an aggregate of more than 75%, the grade is Distinction. If the aggregate is between 60 and 75, the grade is First Division. If the aggregate is between 50 and 60, the grade is Second Division. If the aggregate is between 40 and 50, the grade is Third Division. Otherwise, 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

Enter the marks in Chemistry: 92

Enter the marks in Professional Ethics: 93

Total= 551

Aggregate = 91.83

Class: DISTINCTION

Test Case

Test Case	Inputs
1.	18, 76, 93, 65, 63, 98
2.	73, 78, 79, 75, 87, 0
3.	98, 106, 120, 95, 98, 34
4.	96, 73, -85, 95, 84, 98
5.	78, 59.8, 76, 79, 97, 67

19. Write a program to calculate tax given the following conditions:
    i.   If income is less than or equal to 2,50,000 then no tax
    ii. If taxable income is 2,50,001 – 5,00,000 the charge 10% tax
    iii. If taxable income is 5,00,001 – 10,00,000 the charge 20% tax
    iv. If taxable income is above 10,00,001 then charge 30% tax

    Sample Input:

    Enter the income: 600000

    Sample Output:

    Taxable Income: 350000

    Tax= 35000

Test Case

Test Case	Inputs
1.	400700
2.	2789239
3.	150000
4.	00000
5.	-125486

20. Develop a Java application to generate Electricity bill. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading. Compute the bill amount using the following tariff.
   First 100 units – Rs. 1 per unit
    101-200 units – Rs. 2.50 per unit
    201 -500 units – Rs. 4 per unit
    > 501 units – Rs. 6 per unit

21. Design class called Vehicle, which has two subclasses called Car and Truck. The Vehicle class is the superclass of Car, Truck and Motorcycle. The Vehicle class can contain those fields and methods that all Vehicles need (e.g. a license plate, owner etc.), whereas Car, Truck and Motorcycle can contain the fields and methods that are specific to Car, Truck and Motorcycle. Write a Java program to implement the concept of inheritance based on the following diagram
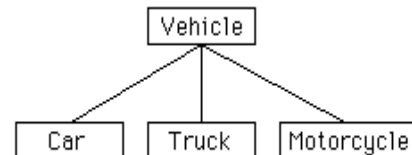
22. Write a program to create a directory that contains the following information.
    (a) Name of a person
    (b) Address
    (c) Telephone Number (if available with STD code)
    (d) Mobile Number (if available)
    (e) Head of the family
    (f) Unique ID No.

23. Write a Java program to create multiple threads for different calculator operations
24. Given a string S partition S such that every substring of the partition is a palindrome. Return the minimum cuts needed for a palindrome partitioning of S

    Input S="aabbc"

    Output=2

    Explanation: The palindrome partitioning ["aa", "bb", "c"]) could be produced using 2 cut.

25. Design a java code an array arr which consists of only zeros and ones, divide the array into three non-empty parts such that all of these parts represent the same binary value.

If it is possible, return any (i, j) with i + 1 < j, such that:

arr(0), arr(1), ..., arr(i) is the first part,

arr(i + 1), arr(i + 2), ..., arr(j - 1) is the second part, and

arr(j), arr(j + 1), ..., arr(arr.length - 1) is the third part.

All three parts have equal binary values.

If it is not possible, return (-1, -1).

Note that the entire part is used when considering what binary value it represents. For example, (1, 1, 0) represents 6 in decimal, not 3. Also, leading zeros are allowed, so (0, 1, 1) and (1, 1) represent the same value.

26. Design a special dictionary that searches the words in it by a prefix and a suffix. Implement the WordFilter class: WordFilter (string () words) Initializes the object with the words in the dictionary. f(string pref, string suff) Returns the index of the word in the dictionary, which has the prefix pref and the suffix suff. If there is more than one valid index, return the largest of them. If there is no such word in the dictionary, return -1.

27. Given an integer array nums, handle multiple queries of the following types: Update the value of an element in nums. Calculate the sum of the elements of nums between indices left and right inclusive where left <= right. Implement the NumArray class: NumArray(int[] nums) Initializes the object with the integer array nums. void update(int index, int val) Updates the value of nums[index] to be val. int sumRange(int left, int right) Returns the sum of the elements of nums between indices left and right inclusive (i.e. nums[left] + nums[left + 1] + ... + nums[right]).
Input
    ["NumArray", "sumRange", "update", "sumRange"]
    ([[1, 3, 5]], [0, 2], [1, 2], [0, 2])
Output
    (null, 9, null, 8)
Explanation
    NumArray numArray = new NumArray((1, 3, 5));
    numArray.sumRange(0, 2); // return 1 + 3 + 5 = 9
    numArray.update(1, 2);   // nums = (1, 2, 5)
    numArray.sumRange(0, 2); // return 1 + 2 + 5 = 8

28. Given an integer array arr, return the number of distinct bitwise ORs of all the non-empty subarrays of arr. The bitwise OR of a subarray is the bitwise OR of each integer in the subarray. The bitwise OR of a subarray of one integer is that integer. A subarray is a contiguous non-empty sequence of elements within an array.
Input: arr = (1, 1, 2)
Output: 3
Explanation: The possible subarrays are (1), (1), (2), (1, 1), (1, 2), (1, 1, 2).
These yield the results 1, 1, 2, 1, 3, 3.
There are 3 unique values, so the answer is 3.

29. Given two numbers arr1 and arr2 in base -2, return the result of adding them together. Each number is given in array format: as an array of 0s and 1s, from most significant bit to least significant bit. For example, arr = (1, 1, 0, 1) represents the number $(-2)^3 + (-2)^2 + (-2)^0 = -3$. A number arr in array, format is also guaranteed to have no leading zeros: either arr == (0) or arr(0) == 1. Return the result of adding arr1 and arr2 in the same format: as an array of 0s and 1s with no leading zeros.

Input: arr1 = [1, 1, 1, 1, 1], arr2 = [1, 0, 1]
Output: [1, 0, 0, 0, 0]
Explanation: arr1 represents 11, arr2 represents 5, the output represents 16.

30. You are given a string of digits num, such as "123456579". We can split it into a Fibonacci-like sequence [123, 456, 579]. Formally, a Fibonacci-like sequence is a list f of non-negative integers such that: 0 <= f[i] < 231, (that is, each integer fits in a 32-bit signed integer type), f.length >= 3, and f[i] + f[i + 1] == f[i + 2] for all 0 <= i < f.length - 2. Note that when splitting the string into pieces, each piece must not have extra leading zeroes, except if the piece is the number 0 itself. Return any Fibonacci-like sequence split from num, or return [] if it cannot be done.

Example-1
Input: num = "1101111"
Output: [11, 0, 11, 11]
Explanation: The output [110, 1, 111] would also be accepted.

Example-2
Input: num = "112358130"
Output: []
Explanation: The task is impossible.

Example-3
Input: num = "0123"
Output: []
Explanation: Leading zeroes are not allowed, so "01", "2", "3" is not valid.