VELTECH RANGARAJAN DR. SAGUNTHALA
R&D INSTITUTE OF SCIENCE AND TECHNOLOGY,
AVADI, CHENNAI. INDIA.

OFFICE OF CAMPUS TO CORPORATE

Problem Solving Questions

Lab Session 1

1. Your task is to create a program that displays the message "Hello world" on the screen. How will you set up your program to print this greeting clearly and correctly?

2. Imagine you have two randomly chosen numbers, and your goal is to write a program that adds them together. How will you set up your program to generate these numbers and calculate their sum?

Lab Session 2

Reading Interactive Input

3. You have two numbers, and your task is to write a program that reads these numbers and calculates their sum. How will you make sure the program adds them together correctly and shows the result?

4. You have three numbers, and your challenge is to write a program that reads these numbers and computes their total sum. How will you ensure your program adds them all together to give the final result?

Lab Session 3

Decision Making

5. Imagine you have two mysterious numbers. Your task is to write a program that reveals which one of them is the bigger one. How will you solve this puzzle?

6. You have two secret numbers, and you need to figure out how they relate to each other using a set of special tools. Your challenge is to write a program that uses these tools—>, >=, <, <=, ==, and !=—to uncover all the secrets about how these numbers compare. How will you use each tool to solve the puzzle?

For example, consider two numbers 15 and 20.

15 < 20 is true.

15 <= 20 is true.

15 > 20 is false.

15 >= 20 is false.

15 == 20 is false.

15 != 20 is true.

7. Imagine you have two secret letters, 'A' and 'B'. Your task is to write a program that uses different comparison tools to uncover how these letters relate to each other. Can you figure out which one is greater or less than the other? Use your program to solve this letter comparison mystery!

For example, consider two characters 'A' and 'B'.

'A' < 'B' is true.

'A' <= 'B' is true.

'A' > 'B' is false.

'A' >= 'B' is false.

'A' == 'B' is false.

'A' != 'B' is true.

8. You have three hidden numbers, and your mission is to find out which one is the greatest. Write a program that can reveal the highest of these three numbers.

8A. Perform the above operation using ternary operator.

9. You have four secret numbers and your challenge is to write a program that figures out which one is the largest and which one is the smallest. Use your trusty if-else statements to solve this number mystery. Can you determine the highest and lowest numbers among them?

Lab Session 4

Divisibility Test

10. You have two numbers, and your task is to write a program that adds them together and checks if their total can be evenly divided by 2. Can you figure out if the sum is even or not?

11. You have a number in hand, and your challenge is to write a program that determines whether this number can be evenly divided by 100. Can you uncover if it's a multiple of 100 or not?

12. You have a number to examine, and your mission is to write a program that checks whether this number can be divided evenly by 27. Can you find out if it's a multiple of 27?

Lab Session 5

13. INFYTQ

You have a year in mind, and your task is to write a program that determines if this year is a leap year. Can you figure out if it has an extra day in February?

A **leap year** is a year that contains one additional day beyond the typical 365 days of a standard year, making it 366 days in total. This extra day is added to the month of February, which normally has 28 days but has 29 days in a leap year.

Key Points About Leap Years:

1. **Purpose:** The extra day compensates for the fact that Earth's orbit around the Sun takes approximately 365.2422 days. Without leap years, our calendar would gradually drift out of sync with the Earth's orbit.
2. **Rules for Leap Years:**
    - **Divisibility by 4:** A year is a leap year if it is divisible by 4. For example, 2020 is a leap year because 2020 ÷ 4 = 505.
    - **Centennial Years:** A year that is divisible by 100 is not a leap year, unless:
        - **Divisibility by 400:** The year is also divisible by 400. For example, the year 2000 was a leap year because 2000 ÷ 400 = 5, while the year 1900 was not a leap year because 1900 ÷ 400 = 4.75.

14. You have two numbers, and your challenge is to write a program that performs both addition and subtraction with them. However, if any subtraction results in a negative number, display it as a positive value. How will you tackle this and show the final results?

For example, consider two numbers 20 and 15.

Addition of 2 values: 20 + 15 = 35.

Subtraction of 2 values: 20 - 15 = 5.

For example, consider two numbers 20 and -150.

Addition of 2 values: 20 + (-150) = -130➔ Absolute value of (-130) = 130.

Subtraction of 2 values: 20 - (-150) = 170.

Lab Session 6

15. TCS 2023.

A washing machine works on the principle of Fuzzy System, the weight of clothes put inside it for washing is uncertain but based on weight measured by sensors and the water level chosen, it decides total time needed.

For low level water, the time estimate is 25 minutes, where approximately weight is between 2000 grams or any nonzero positive number below that.

For medium level water, the time estimate is 35 minutes, where approximately weight is between 2001 grams and 4000 grams.

For high level water, the time estimate is 45 minutes, where approximately weight is above 4000 grams. Assume the capacity of machine is maximum 7000 grams.

When the weight is zero, time needed is 0 minutes.
If the weight exceeds maximum weight limit, then, print "OVERLOADED", and for all negative weights, the output is "INVALID INPUT".

Sample Input-1: 2000, L

Sample Output-1: Time Estimated: 25 minutes

Input should be in the form of integer value.

Output must have the following format: Time Estimated: NN Minutes

Switch Case

16. Imagine you're building a handy calculator with five basic operations: addition (+), subtraction (-), multiplication (*), division (/), and modulus (%). Your task is to write a program that can perform any of these operations based on user input. How will you create this versatile calculator to solve various arithmetic problems?

16A. Write a program to execute switch case without break statement *(Java stream only).*

17. Write a program to implement calculator program with 5 basic arithmetic operations with input choice as a character (e.g. use 'a' for addition, 's' for subtraction, 'm' for multiplication, 'd' for division and 'u' for modulus operation, which gives the reminder of the division).

For example, consider two number 20 and 15 and an input 'a'.

'a' means addition, so, 20a15 = 20 + 15 = 35.

For example, consider two number 20 and 15 and an input ‘u’.

‘u’ means modulus, so, 20u15 = 20 mod 15 =5.

18. Create a program that acts as a calculator capable of handling five basic arithmetic operations. You’ll use the symbols `+` for addition, `-` for subtraction, `*` for multiplication, `/` for division, and `%` for finding the remainder. How will you design your program to perform these operations based on user input?

Lab Session 7

Loops

19. Imagine you’re on a quest to discover never-ending loops. Create a program that demonstrates two types of endless journeys: one using a `for` loop and another using a `while` loop. How will you set up these loops to keep running forever, showcasing their infinite nature?

20. Imagine you need to repeat a cheerful message. Write a program that uses a `for` loop to print "ALL IS WELL" exactly twenty times. How will you set up your loop to ensure this message appears the right number of times?

21. You have a number (n), and your challenge is to write a program that prints out a sequence starting from 1 and going up to (n). How will you design your program to generate and display this sequence of numbers?

22. You need to write a program that reads a number (n) and prints all numbers from 1 up to (n). However, there’s a twist: keep the initialization part outside of the `for` loop. How will you structure your program to accomplish this?

Sample input:

Assume that N =5.

Sample output:

The first 5 natural numbers are 1, 2, 3, 4, 5.

23. You need to write a program that reads a number N and prints all numbers from 1 up to N. The challenge is to keep the initialization statement outside of the for loop and place the increment or decrement as the last statement inside the for loop body. How will you design your program to meet these conditions and produce the desired sequence?

Sample input:

Assume that N =5.

Sample output:

The first 5 natural numbers are 1, 2, 3, 4, 5.

24. You have a number (n), and your task is to write a program that prints the squares and cubes of all numbers from 1 up to n. Use a `while` loop to generate and display these values. How will you set up your program to calculate and show both the squares and cubes for each number in the range?

Sample input:

Assume that N =5.

Sample output:

The square of first 5 natural numbers are 1, 4, 9, 16, 25.

The cube of first 5 natural numbers are 1, 8, 27, 64, 125.

25. You need to write a program that reads a number (n) and calculates the sum of the first (n) natural numbers. Use any type of loop to accomplish this. How will you design your program to sum these numbers and display the result?

Sample input:

Assume that n =5.

Sample output:

Sum of first 5 natural numbers = 1 + 2 + 3 + 4 +5 = 15.

Lab Session 9

25A. ***(Java stream only)***

Write a program to read print the "welcome to the do while loop" greeting message using do while loop. Just get the number n as input and print the message n times.

25B ***(Java stream only)***

Imagine you're building a calculator that can perform different arithmetic operations. Write a program that uses a `do-while` loop and a `switch` statement to handle user input and perform calculations. How will you set up your program to repeatedly ask for input and execute operations like addition, subtraction, multiplication, and division based on user choice?

25C. ***(Java stream only)***

You're tasked with creating a simple calculator using a `while` loop and a `switch` statement. Your program should repeatedly prompt the user to choose an arithmetic operation (like addition, subtraction, multiplication, or division) and then perform the selected operation based on user input. How will you set up your `while` loop and `switch` case to keep the calculator running until the user decides to exit?

Lab Session 10, 11

Java method / User defined functions -- Discussion

26A. Write a program to demonstrate functions of Type 1- that is function without return type & without parameter.

26B. Write a program to demonstrate functions of Type 2 - that is function without return type & with parameter.

26C. Write a program to demonstrate functions of Type 3 - that is function with return type & without parameter.

26D. Write a program to demonstrate functions of Type 4 – that is function with return type & with parameter.

Working with Function

27. Generation of odd series.

Create a method that prints the first (n) odd numbers, where (n) is provided as input. How will you design this method to generate and display the sequence of odd numbers correctly?

```
void oddSeries (int n) {

}
```

Sample Input/ Output:

Input: n = 5
Output: 1 3 5 7 9

28. Generation of even series:

Create a method that prints the first (n) even numbers, where (n) is the input. How will you design this method to generate and display the sequence of even numbers correctly?

```
void evenSeries (int n) {

}
```

Sample Input/ Output:

Input: n = 5
Output: 2 4 6 8 10

29. Generation of Table:

Create a method that prints the multiplication table for a given number (n). The table should include the first 10 terms. How will you design this method to generate and display the multiplication table?

```
void printTable (int n) {

}
```

Sample Input/ Output:

Input:

n = 5

Output:

5 * 1 =5
5 * 2 =10
5 * 3 =15
5 * 4 =20

5 * 5 =25
5 * 6 =30
5 * 7 =35
5 * 8 =40
5 * 9 =45
5 * 10 =50

30. Generation of Square Table

Create a method that prints the squares of all numbers from 1 up to a given number (n). How will you design this method to generate and display the squares of numbers in this range?

```
void printSquareTable (int n) {

}
```

Sample Input/ Output:
Input:
n = 5
Output:
Square (1) =1
Square (2) =2
Square (3) =9
Square (4) =16
Square (5) =25

31. Generation of Cube Table:

Create a method that prints the cubes of all numbers from 1 up to a given number (n). How will you design this method to generate and display the cubes of numbers within this range?

```
void printCubeTable (int n) {

}
```

Input:
n = 5
Output:
Cube (1) =1
Cube (2) =8
Cube (3) =27
Cube (4) =64
Cube (5) =125

32. Generation of Fibonacci series. You need to write a method that prints the first (n) numbers in the Fibonacci series, where (n) is provided as input. How will you design this method to generate and display the sequence of Fibonacci numbers up to (n)?

Note: Fibonacci series: 1, 1, 2, 3, 5, 8, 13, 21 .... We get the next Fibonacci number, by adding the previous two Fibonacci numbers.

Recursive Formula to find the nth Fibonacci number:

fib (n) = fib (n-1) + fib(n-2), if n > 2
fib (1) = 1
fib (2) = 1

Note: Non-recursive solution is faster.

```
void fib (int n) {
        // write code
}
```

33. You have a number and need to check if it belongs to the Fibonacci sequence, where each number is the sum of the two preceding ones. Write a method to determine if this number is part of the Fibonacci series. Return `true` if it is, and `false` if it isn't. How will you uncover whether this number fits into the famous Fibonacci pattern?

```
boolean isFib (int n) {
    //write code
}
```

Hint: Here's a step-by-step approach to checking if a number is a Fibonacci number:

1. **Calculate Two Values:**
   - Compute 5*n^2 + 4.
   - Compute 5*n^2 − 4.
2. **Check for Perfect Squares:**
   - Check if either 5n^2+4 or 5n2−4 is a perfect square.
   - **If it is a perfect square, then, given number is a Fibonacci number.**

Sample Input/ Output 1:

Input: n = 5
Output: True. 5 is a Fibonacci number.

Sample Input/ Output 2:

Input: n = 15

Output: False. 15 is not a Fibonacci number.

34. Imagine you've stumbled upon a magical sequence known as the Fibonacci series. Your task is to uncover the (n)th number in this intriguing sequence. Write a method to find and reveal the number that sits at position (n) in the Fibonacci series. Can you unravel this sequence and discover the (n)th term?

Note:

Fibonacci series:	1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...
Nth Fibonacci number index:	1 2 3 4 5 6 7 8 9 10

Sample Input/ Output 1:

Input: n = 5

Output: 5th Fibonacci number =5.

Sample Input/ Output 2:

Input: n = 10

Output: 10th Fibonacci number =55.

35. CAPGEMINI 2023

Write a method to solve the following equation $a^3 + a^2b + 2a^2b + 2ab^2 + ab^2 + b^3$. Write a program to accept three values in order of a, b and c and get the result of the above equation.

```
int findValue (int a, int b, int c) {

}
```

Hint: Use Math.pow (x, y) for finding the x value raised to the power of y. But, use the type casting (int) to get the integer result from the obtained power.

int k = (**int**) Math.pow (x, y); // this will avoid the compile time error.

Lab Session 14, 15

36. CAPGEMINI 2023

A method is there which tells how many dealerships are there and the total number of cars and number of two wheelers in each dealership. Your job is to calculate how many tyres would be there in each dealership and find the total number of tyres.

```
int findValue (int n, int a [], int b [])    {

}
```

Input

n=3
Dealership 1: number of cars = 4, number of two wheelers = 2
Dealership 2: number of cars = 4, number of two wheelers = 0
Dealership 3: number of cars = 1, number of two wheelers = 2

Output: 44

Explanation: There are total 3 dealerships: Dealerships1 contains 4 cars and 2 bikes. Dealerships2 contains 4 cars and 0 bikes. Dealerships3 contains 1 car and 2 bikes

Total number of tyres needed in dealerships1 is (4 x 4) + (2 x 2) = 20
Total number of tyres needed in dealerships2 is (4 x 4) + (0 x 2) = 16
Total number of tyres needed in dealerships3 is (1 x 4) + (2 x 2) = 08.
Total tyres = 20+16+8 = 44.

37) You have a number and need to find out how many digits it contains. How will you design your method to determine the total count of digits accurately?

```
int countNumberOfDigits (int input1) {
    // input is passed in "input1" argument
    // write the logic
    // return answer
}
```

Sample Input/output 1:
Consider input1=12377.
Expected answer = 5.
Explanation: there are 5 digits' in the given number.

Sample Input/output 2:

Consider input1=77.
Expected answer = 2.
Explanation: there are 2 digits in the given number.

38) You have a number, and your challenge is to find the sum of its digits. Write a method that adds up each digit in the number and gives you the total. How will you craft your method to calculate this sum effectively?

```
int sumOfDigits (int input1) {
        // write the logic
        // return answer
}
```

Sample Input / Output 1:

Consider input1=12377. Expected answer = 20.
Explanation: 1+2+3+7+7 = 20.

39) You have a number, and your task is to repeatedly sum its digits until you obtain a single-digit result. Write a method that performs this process and returns the final single-digit sum. How will you design your method to handle the iterative summing and achieve this ultimate single-digit result?

```
int sumUptoSingleDigit (int input1) {
                        // write the logic and return answer
}
```

Sample Input/output 1:

Consider input1=12377.
Expected answer = 2.
Explanation: 1+2+3+7+7 = 20. Now, 20 is a two-digit number. So, again apply the given logic on this number 20. Now, 2+0 =2. 2 is a single digit number. So, stop and return the answer.

Sample Input/output 2:

Consider input1=9999.
Expected answer = 9.
Explanation: 9+9+9+9 = 36. Now, 36 is a two-digit number. So, again apply the same logic on this number 36. Now, 3+6 =9. 9 is a single digit number. So, stop and return the answer.

Lab Session 16

40) You have a number (n) and need to unravel its factorial. Write a method that calculates the factorial of this number, which is the product of all positive integers up to (n). Can you figure out how to find this magical product?

```
int fact (int n) {
        // CODE
}
```

Note: 0! = 1; 1! = 1

Verify the answers, by changing the input values.

2! = 2; 3! = 6; 4! = 24; 5! = 120.

41) Imagine you have two numbers, (n) and (r), and you need to uncover the secret value of (nCr), which represents the number of ways to choose (r) items from (n) items without regard to order. Write a method to calculate this value. How will you solve this combinatorial puzzle?

Hint: nCr (n, r) = n! / ((n-r)! *r!)

```
int nCr (int n, int r) {

        // write code
}
```

Sample Input/Output-1:

nCr (5,2) --> 5! / ((3!) *(2!)) =10

Sample Input/Output-2:

nCr (5,1) --> 5! / ((4!) *(1!)) =5

42) You have a number, and you need to find the sum of the digits that are prime numbers (2, 3, 5, or 7). Write a method to extract these prime digits and calculate their sum. How will you design your method to identify and total these prime digits from the given number?

```
int primeDigitSum (int input1) {
                        // write the logic and return answer
}
```

Sample Input / Output 1:

Consider input1=12377.

Expected answer = 19.

Explanation: 2+3+7+7 =19.We need to sum "only" the prime digits. Note that, 1 is neither prime nor composite. So, ignore 1.

43) You have a number and need to determine if it is a prime number, which means it has no divisors other than 1 and itself. Write a method that tests whether this number is prime. How will you design your method to uncover if the number is truly a prime?

```
boolean isPrime (int N) {
        // write code
}
```

Explanation: A number is prime, if it has **exactly** 2 factors, which are 1 and the number itself. Prime numbers have 1 and itself alone as factors.
Consider N =5. From 1 up to 5, inclusive of 5, we divide:

1 divides 5.
5 divides 5.
no other number divides 5.
So, number of factors = 2 → (which are 1 and 5). Hence, 5 is PRIME number.

Consider N = 121; Factors of 121 are 1,11,121.
So, number of factors = 3 → (which are 1 ,11, and 121). Hence, 121 is a composite number.

Sample Input/Output-1:
isPrime (17) → true because 17 is a prime number.

Sample Input/Output-2:
isPrime (15) → false because 15 is not a prime number.

44) WIPRO 2023 - The Prime Range Riddle

In the kingdom of numbers, you are given a range of integers and need to determine how many of them are prime. Your quest is to count all the prime numbers that lie within this specified range.

The Challenge:

1. **Receive the Range:** You are provided with two integers, start and end, which define the inclusive range [start, end].
2. **Identify Prime Numbers:** Determine which numbers within this range are prime.
3. **Count the Primes:** Calculate the total number of prime numbers in this range.
4. **Return the Count:** Provide the number of primes found.

Rules to Follow:

- A **prime number** is a natural number greater than 1 that has no positive divisors other than 1 and itself.
- For example, in the range [10, 50], the prime numbers are 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47.

Example:

- **Given Range:** 10 to 50
  - **Primes in this Range:** 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47
  - **Total Count of Primes:** 10

Can you solve the Prime Range Riddle and find out how many prime numbers lie between the given start and end values?

45) You have a number and need to determine if it is a Strong number. A Strong number is one where the sum of the factorials of its digits equals the number itself. Write a method to check this property. How will you design your method to compute the factorials of the digits, sum them up, and verify if the total matches the original number?

```
boolean isStrong (int n) {
        // write code
}
```

Sample Input/output 1:

Consider n=124.
Individual digits are 1, 2, 4.
Let sum = (1!) + (2!) + (4!) = 1+2+24 = 27.
Here, given number n= 124. That is, sum of factorial of digits is NOT equal to given number n. So, 124 is **NOT** a Strong Number.

Sample Input/output 2:
Consider n=145.
Individual digits are 1, 4, 5.
=> (1!) + (4!) + (5!) => 1+24+120= 125.
Here, Sum of factorial of digits is equal to given n. So, 145 is a Strong Number.

Lab Session 18, 19

46) You have a number and need to check if it is an Armstrong number.
An Armstrong number is one where the sum of the cubes of its digits equals the number itself. Write a method to test this property.
How will you design your method to compute the cubes of the digits, add them up, and confirm if the total matches the original number?

```
boolean isArmstrong (int N) {
        // write code
}
```

Sample Input/output 1:
Consider n=153. Number of digits in the given number = 3.
Individual digits are 1, 5, 3.
Sum = cube (1) + cube (5) + cube (3)
Sum = 1+125+27 = 153.
Here, Sum of cube of digits is equals to given number n. So, 153 is an Armstrong Number.

Sample Input/output 2:
Consider n=123. Individual digits are 1, 2, 3.
==> cube (1) + cube (2) + cube (3)
==> 1+8+27 = 36.
Sum of cube of digits is NOT equal to given n. So, 123 is not an Armstrong number.

47) You need to write a method that generates a range of numbers starting from a given value and ending at another, incrementing by a specified step value. Your method should return all these numbers as an array of integers. How will you design your method to create and return this sequence of numbers based on the start, end, and step values?

```
int [] range (int start, int end, int step) {
        // CODE
}
```

Note: The actual ending value, *end*, will not to be included in the output. This is similar to **range function in python**.

Sample Input/Output-1:

input (start=1, end=10, step=1)
answer = {1,2,3,4,5,6,7,8,9}

Sample Input/Output-2:

input (start=0, end=11, step=2)
answer = {0,2,4,6,8,10}

48. TCS 2023:

Given a maximum of four digits to the **base 17**, as input, output its decimal value. Note that, 10 is represented as A, 11 is represented as B, 12 is represented as C, F is represented as 15, 16 is represented as G.

Sample Input : 23GF

Sample Output : 10980

Explanation:

Digits→	**2**	**3**	**G is 16**	**F is 15**
Position from RHS:	3	2	1	0
Power of 17:	17^3	17^2	17^1	17^0
Value	**2***4913	3*289	16*17	15*1
Total Value	**(2*4913)+ (3*289)+ (16*17) + 15 = 10980**			

Lab Session 20

49. ACCENTURE 2023

N-base notation is a system for writing numbers that uses only n different symbol. The first n symbols from the given notation has to be used (Including the symbol for o)

Decimal to n base notation are (0:0, 1:1, 2:2, 3:3, 4:4, 5:5, 6:6, 7:7, 8:8, 9:9, 10: A,11: B and so on up to 35: Z).

Implement the following method **char [] decimalToBaseN (int n, int number)**: The method accepts positive integer n and num. Implement the method to calculate the n-base equivalent of number and return the same as a string.

Steps:

- Divide the decimal number by n, Treat the division as the integer division
- Write the remainder (in n-base notation)
- Divide the quotient again by n, Treat the division as integer division
- Repeat step 2 and 3 until the quotient is 0
- The n-base value is the sequence of the remainders from last to first

Assumption: 1 < base < = 36

Example Input:

12→ base

718 → n

Output

4BA

Explanation

Iteration Number	Given number	Given Divisor	Quotient	Remainder
1	718	12	59	10 which is coded as A
2	59	12	4	11 which is coded as B
3	4	12	0	4 which is coded as 4

50. ACCENTURE 2023

A carry is a digit that is transferred to left if sum of digits exceeds 9 while adding two numbers from right-to-left one digit at a time. You are required to implement the following method.

int numberOfCarries (int num1, int num2);

The methods accept two numbers ‘num1’ and ‘num2’ as its arguments. You are required to calculate and return the total number of carries generated while adding digits of two numbers ‘num1’ and ‘num2’.

Assumption: num1, num2>=0

Carry		0 1 1
number1	=	4 5 1
number2	=	3 4 9
sum	=	8 0 0

Output 1: 2

Explanation:

Adding ‘num 1’ and ‘num 2’ right-to-left results in 2 carries since (1+9) is 10. 1 is carried and (5+4=1) is 10, again 1 is carried. Hence 2 is returned.

Sample Input2:

Number 1: 23

Number 2: 563

Sample Output2: 0 (as there are no carries generated).

53. Create a method that determines if a given number is odd or even. If the number is odd, the method should return "odd"; if it's even, it should return "even". How will you write this method to check the number and provide the correct result?

```
String isOddOrEven (int n) {
        // write code
}
```

Sample Input/ Output:

Input: n = 10

Output: even

54. Create a method that checks if a given number (N) is a multiple of another number (X). The method should return `true` if (N) is indeed a multiple of (X), and `false` if it is not. How will you design this method to perform the check and return the appropriate result?

```
boolean isMulOfX (int n, int x) {
        // return true, if N is a multiple of X
        // return false, if N is NOT a multiple of X

 }
```

Sample Input/Output-1:
isMulOfX (10,2) --> true. Because, 10 is a multiple of 2.

Sample Input/Output-2:
isMulOfX (10,3) --> false. Because, 10 not is a multiple of 3.

Sample Input/Output-3:
isMulOfX (10,10) --> true. Because, 10 is a multiple of 10.

55. Create a method that calculates the area of a square, given the side length, a. If the length is negative, the method should return -1. How will you implement this method to handle both positive and negative values and compute the area correctly?

```
float areaOfSquare (int a) {
    // code
}
```

Sample Input/ Output:
Input: a = 10
Output: 100

Lab Session 22

56. Create a method to calculate the volume of a cube, where the side length of the cube is given by (a). If (a) is negative, the method should return -1. How will you design this method to handle the calculation and deal with negative input values?

```
float volumeOfCube (int a) {
    // code
}
```

Sample Input/ Output:
Input: a = 10
Output: 1000

57. Create a method that calculates the area of a rectangle using its length and breadth. If either the length or breadth is negative, the method should return -1. How will you design this method to perform the area calculation and handle negative inputs appropriately?

```
float areaOfRectangle (float length, float breadth) {
    // code
}
```

Sample Input/ Output:
Input:
l = 10
b = 20
Output: 200

58. Create a method to calculate the area of a circle given its radius. If the radius is negative, the method should return -1. How will you write this method to compute the area correctly while handling negative radius values?

Hint: Area of Circle = PI*r*r, where PI = 3.14 and r is the given radius.

```
float areaOfCircle (float radius) {
    // code
}
```

Sample Input/ Output:
Input: r = 10
Output: 314

59. Create a method that calculates both simple interest and compound interest given the principal (p), the number of years (n), and the rate of interest (r). If any of these values are negative, the method should return -1. How will you implement this method to compute the interests accurately and handle invalid inputs?

```
float findSI (float principal, float number_of_years, float rateOfInterest) {
    // write code
}

float findCI (float p, float n, float r) {
    // write code
}
```

Hint:
Simple interest = (P *N * R) / 100
Compound Interest =(P*(1+R/100)) ^N − P
In java: Use Math.pow (x, y) to calculate the x^y.

Sample Input/ Output:
Input:
p = 1000
n = 1

r = 10

Output:

Simple Interest (SI) = (p * n* r) / 100 = (1000*1*10) / 100 = 100
Compound Interest (CI) = (1000 * (1+10/100)) ^1 − 1000 = 100

Note: SI and CI for the first year will be same.

Lab Session 23, 24

PATTERN PRINTING USING NESTED LOOPS

60. Write a program to understand nested loops. Print the letters of the word as shown below.

Example:

Sample input as “possible”

Sample output: p oo sss ssss iiiii bbbbbb lllllll eeeeeeee.

Explanation:

The ith character is printed ‘i’ times.

That is, first character ‘p’ is printed **one** time.

And, second character ’o’ is printed **two** times and so on.

61. CAPGEMINI 2023

You’re supposed to reduce the size of this string using mathematical logic given as in the example below:

Sample Input-1: aabbbbeeeeffggg
Sample Output -1: a2b4e4f2g3

Sample Input -2: abbccccc
Sample Output- 2: ab2c5

62. CAPGEMINI 2023

You have write a method that accepts, a string which length is “len”, the string has some “#”, in it you have to move all the hashes to the front of the string and return the whole string back and print it.

Method Syntax: **String moveHash (String str);**
Sample input 1: Move#Hash#to#Front

Sample Output: ###MoveHashtoFront

63. Write a program to print the following pattern, given a number, using 'nested for loops'.

Sample input: 3
Sample output:

```
* * *
* * *
* * *
```

64 Write a program to read a number, and print the following pattern using 'nested for loops'.

Sample input: 3
Sample output:

```
*
* *
* * *
```

65. Given a number as input, print the following pattern using 'nested while' loops.

Example: INPUT: N=5

OUTPUT:

```
*
* A*
* B * *
* C * * *
* D * * * *
```

66. Given n as input, print the following pattern using 'nested for loops'

Example: INPUT: N=5

OUTPUT (print the **Z** like pattern using #):

```
# # # # #
      #
    #
  #
# # # # #
```

Note that, the output totally has 5 rows and first row and last row are having same number of hashes.

67. Write a method to print the following pattern.

```
void printPattern2 (int input1) {
        // CODE
}
```

Sample Input/output 1: Consider input1 is 4.
Expected answer:

```
1
1 3
1 3 5
1 3 5 7
```

68. Write a method to print the following pattern, which looks a banner format of letter "C".

```
void print_C_BannerStars (int input1) {
        // CODE
}
```

Sample Input / Output 1: Consider input1 is 4.
Expected answer:

```
* * * *
* *
* *
* * * *
```

Sample Input / Output 2: Consider input1 is 5.
Expected answer:

```
* * * * *
* *
* *
* *
* * * * *
```

69. Write a method to print the following pattern.

```
void print1Pattern1(int input1) {
        // CODE
}
```

Sample Input/output 1: Consider input1 is 4.
Expected output:

```
1 * * * * * * * * 1          #8stars
1 2 * * * * * * 2 1          #6stars
1 2 3 * * * * 3 2 1          #4stars
1 2 3 4 * * 4 3 3 1          #2stars
```

70. Write a method to print the following pattern.

```
void printStarNumberHash (int input1) {
        // CODE
}
```

Sample Input/output 1:
Consider input1 is 5.
Sample output:

```
  * 1 # # *
  * * 2 2 $ $ * *
  * * * 3 3 3 # # * * *
  * * * * 4 4 4 4 $ $ * * * * *
  * * * * * 5 5 5 5 5 # # * * * * *
```

71) Write a method to print the following pattern. The given number n represents the number of rows in the output.

```
void printStarsPattern2 (int input1) {
        // CODE
}
```

Sample Input/output 1:
Consider input1 is 4.
Expected answer:

```
      *
     * *
    * * *
   * * * *
```

Lab Session 27

One Dimensional Arrays

72) You need to write an efficient method to find the minimum value in a given array, but without using any `if` statements. How will you design your method to determine the smallest value in the array using alternative approaches, such as built-in functions or mathematical operations?

```
int findMin (int data []) {
        // write code
}
```

Sample input/ output:

The function **findMin** efficiently finds the smallest number in an array by comparing each element and updating the minimum value accordingly. In the given example, the minimum value in the array {34, 7, 23, 32, 5, 62, 29} is **5**.

73) You need to write an efficient method to find the maximum value in a given array without using `if` statements. How will you craft your method to determine the highest value in the array using alternative techniques, such as built-in functions or mathematical operations?

```
int findMax (int data []) {
        // write code
}
```

Sample input/ output:

The findMax function efficiently determines the largest number in an array by comparing each element and updating the maximum value accordingly. In the provided example, the maximum value in the array {34, 7, 23, 32, 5, 62, 29} is 62.

74) You need to write an efficient method to calculate the sum of all values in a given array. How will you design your method to quickly and accurately compute the total of the array's elements?

```
int findSum (int data []) {
        // write code
}
```

Sample Input/output 1:

input: {1, 2, 3, 4}

output: sum of all elements in array = 10

75) You need to write a method that takes an array and reverses its order. The method should return the reversed array. How will you design your method to flip the elements in the array and provide the reversed version?

```
int [] reverseArray (int [] array)        {
        // CODE
}
```

Sample Input/output 1:

Input: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
Output: {10, 9, 8, 7, 6, 5, 4, 3, 2, 1}

Lab Session 28

76) You need to write a method that searches for a value (x) in a sorted array. If (x) is present in the array, the method should return `true`; otherwise, it should return `false`. How will you design your method to efficiently find (x) in the sorted array and determine its presence?

```
boolean binarysearch (int array [], int x)      {
        // write code
}
```

Sample Input/output 1:
Consider the sorted array [] = {5, 10, 15, 40, 45, 65}

- binarysearch (array, 40) should return true. Because, 40 is found in the array.
- binarysearch (array, 43) should return false. Because, 43 is NOT found in the array.

Sample Input/output 2:
Consider the sorted array [] = {15, 20, 25}

- binarysearch (array, 40) should return false. Because, 40 is NOT found in the array.
- binarysearch (array, 43) should return false. Because, 43 is NOT found in the array.

76A) Make all array elements equal just by using decrement by 1 operation or increment by 1 operation. You need to make all the values equal to the minimum value in the given array. And, return the total number of operations needed to make all the elements equal as per the given description.

Note: Return the answer, without actually performing the above operation.

```
int makeAllValuesEqual (int a []) {
        //write code
}
```

Sample Input/output 1:

Input: {1, 2, 3}
Output: Minimum value in the array = 1
Total number of operations = absolute (2-1) + absolute (3-1) = 1+2 = 3.

Lab Session 29

77) Write a method to find the maximal subarray sum. Kadane's Algorithm is an efficient method to find the maximum sum of a contiguous subarray within a one-dimensional array of numbers. It's widely used in computer science and competitive programming because of its simplicity and efficiency. The algorithm operates in linear time, O(n), where n is the number of elements in the array.

```
int findMaxSubArraySum (int a []) {
        //write code
}
```

Sample Input/output 1:

Input Array: {5, 10, -15, 20, -30}
Output: 20.

Explanation:

Consider subarray with first 2 numbers. 5+10 =15.
Consider subarray with first 3 numbers. 5+10+ (-15) = 0.
Consider subarray with first 4 numbers. 5+10+ (-15) + 20 =20.
Consider subarray with first 5 numbers. 5+10+ (-15) + 20 + (-30) = -10.

In this case, an array consisting of first four numbers gives the maximum sub array sum.

78. CAPEGEMINI 2023

You're given with the size of the array and an array of integers; print the number of times each integer has occurred in the array.

Sample Input 1:

10

1 2 3 3 4 1 4 5 1 2

Sample Output 1:

1 occurs 3 times
2 occurs 2 times
3 occurs 2 times
4 occurs 2 times
5 occurs 1 times

79. Write a method to find the number of longest increasing sub sequences (LIS) found in the given array and return the same.

- Longest Increasing Subsequence (LIS) finds the longest subsequence where elements strictly increase in order.
- The LIS problem involves finding the longest subsequence of a sequence where each element is greater than the previous one, preserving order.

```
int countOfLIS (int data []) {
        //write code
}
```

Sample Input 1: 1 1 2 3 3 4 1 4 5 1 2

Sample Output 1: Number of longest increasing sub sequences = 3

(Sub sequence 1: 1 1 2 3 3 4, sub sequence 2: 1 4 5, Sub sequence 3:1 2)

Lab Session 30

80. Write a method to find the number of longest decreasing sub sequence (LDS) found in the given array and return the same. The LDS problem seeks the longest subsequence where each element is smaller than the preceding one, while maintaining the original order.

```
int countOfLDS (int data []) {
        //write code
}
```

Sample Input 1:

11 1 -2 -3 13 4 1 14 5 1 -2

Sample Output 1:

Number of longest decreasing sub sequences = 3

(Sub sequence #1: 11 1 -2 -3, sub sequence #2: 13 4 1, Sub sequence #3: 14 5 1 -2)

81. Write a method to remove all repeated elements from an array. The resultant array should have only unique elements.

```
int [] removeDuplicates (int [] array)          {
            // CODE
}
```

Sample Input/output 1:

Input Array: {5, 10, 15, 20, 5, 30}
Output Array: {5, 10, 15, 20, 30}.
In this case, number 5 is **repeated**. So, it is removed.

Sample Input/output 2:

Input Array: {5, 10, 15}
Output Array: {5, 10, 15}.
In this case, all given numbers are **unique.** Hence, no number is removed from the array.

Lab Session 31

82. ACCENTURE 2023

The method accepts two positive integers ‘r’ and ‘unit’ and a positive integer array ‘arr’ of size ‘n’ as its argument ‘r’ represents the number of rats present in an area, ‘unit’ is the amount of food each rat consumes and each ith element of array ‘arr’ represents the amount of food present in ‘i+1’ house number, where 0 <= i.

Note: Return -1 if the array is null
Return 0 if the total amount of food from all houses is not sufficient for all the rats.

Sample Input:

7 – Number of rats
2 – Food needed for each rat
8 – Number of houses
{2, 8, 3, 5, 7, 4, 1, 2} – Food available in each house

Sample Output: 4
Explanation:
Total amount of food required for all rats = r * unit; which is = 7 * 2 = 14. The amount of food in 1st houses = 2+8+3+5 = 18. Since, amount of food in 1st 4 houses is sufficient for all the rats. That is, 4 houses to be visited to satisfy all rat’s food need. Thus, output is 4.

83. TCS 2023

A party has been organised on cruise. The party is organised for a limited time(T). The number of guests entering (E[i]) and leaving (L[i]) the party at every hour is represented as elements of the array. The task is to find the maximum number of guests present on the cruise at any given instance within T hours.

Sample Input 1: T=5

Hour →	1	2	3	4	5
Number of people coming into the party →	7	0	5	1	3
Number of people going out of the party →	1	2	1	3	4

Sample Output :8

Explanation:

1st hour: Entry: 7 Exit: 1
No. of guests on ship: 6
2nd hour: Entry: 0 Exit: 2
No. of guests on ship: 6-2=4
Hour 3: Entry: 5 Exit: 1
No. of guests on ship: 4+5-1=8
Hour 4: Entry: 1 Exit: 3
No. of guests on ship: 8+1-3=6
Hour 5: Entry: 3 Exit: 4
No. of guests on ship: 6+3-4=5
Hence, the maximum number of guests within 5 hours is 8.

The input format for testing

The candidate has to write the code to accept 3 input.

First input- Accept value for number of T (Positive integer number)
Second input- Accept T number of values, where each value is separated by a new line.
Third input- Accept T number of values, where each value is separated by a new line.

The output format for testing

The output should be a positive integer number or a message as given in the problem statement (Check the output in Example 1)

Two Dimensional Arrays

84. Write a method to find sum of two matrices and return the resulting matrix.

```
int [] [] addMatrix (int a [] [], int b [] []) {
        // write code
}
```

Sample Input:

- Matrix A:

  1 2 3
  4 5 6
  7 8 9

- Matrix B:

  9 8 7
  6 5 4
  3 2 1

Sample Output:

The resulting matrix after adding A and B will be:

10 10 10
10 10 10
10 10 10

85. Write a method to multiply two matrices and return the resulting matrix.

```
int [] [] multiplyMatrix (int a [] [], int b [] []) {
        // write code
}
```

Sample Input:

Matrix A:

1 2
3 4

Matrix B:

5 6
7 8

Sample Output:

Resulting matrix = 19 22

43 50

86. Write a method to perform given operation on all elements of the matrix.
Modify entire matrix, using the given x value, based on operator given operator. Input operator can be '+' or '-' or '*' or '/' or '%'.

```
int [] [] modifyMatrix (int a [] [], char operator, int x) {
        //write code
}
```

Sample Input: operator = '+' , x = 2

Matrix A:

1 2
3 4

Sample Output:

Resulting matrix:

1+2=3 2+2=4
3+2=5 4+2=6

String Handling

87. Write a program to print ASCII values of all characters in the given array.

```
void printASCII( int data[]) {

}
```

Consider the following array and performing the above operation.

Sample input:

```
char [] name = {‘a’, ‘A’, ‘z’, ’Z’, ‘0’};
```

Sample output:

```
-----------------------------------
 char                ASCII Value
-----------------------------------
a                    97
A                    65
z                    122
Z                    90
0                    48
-----------------------------------
```

88. You need to write a method that calculates the sum of the ASCII values of all characters in a given string. How will you design your method to iterate through the string and compute the total ASCII sum for its characters?

```
int asciiSum (String s) {
        // write code
}
```

Sample Input/Output-1:

asciiSum("AA") → 130

Sample Input/Output-2:

asciiSum("aa") → 97+97=194

89. You need to write a method that converts all characters in a given array to uppercase. How will you design your method to transform each character in the array to its uppercase equivalent?

Sample input: char [] name = {‘a’, ‘A’, ‘z’, ’Z’};

Sample output: char [] name = {‘A’, ‘A’, ‘Z’, ’Z’};

90. You need to write a method that counts and prints the number of vowels present in a given array of characters.

How will you design your method to identify vowels and tally their occurrences?

Sample input:

char [] name = {‘a’, ‘A’, ‘z’, ’Z’, ‘0’};

Sample output:

Number of vowels = 2

Lab Session 34, 35

91. You need to write a method that identifies and prints the unique consonants present in a given array of characters.

How will you design your method to filter out and display each distinct consonant found in the array?

Sample input:

char [] name = {‘a’, ‘A’, ‘z’, ’Z’};

Sample output:

Number of unique consonants = 2

92. You need to write a method that counts and prints the number of lower case consonants and upper case vowels in a given array of characters.

How will you design your method to accurately tally and display the counts for both consonants and vowels?

Sample input:char [] name = {‘a’, ‘A’, ‘z’, ’Z’};

Sample output:

Number of lower case consonants = 2

Number of upper case vowels = 1

93. You need to write a method that counts and prints the number of vowels and consonants in the given string, considering case sensitivity. How will you design your method to accurately identify and count the vowels and consonants in this specific string?

Sample input: String name = “VEL”;

Sample output: Number of vowels = 1
Number of consonants = 2

94. You need to write a method that converts a given string into a character array and then prints out the array, where each character is separated by comma. How will you design your method to perform the conversion and display the characters in the array?

Sample input:

String name = “VEL”;

Sample output:
Characters in the given array: V, E, L

95. You need to write a method that converts all the letters in a given string to uppercase. How will you design your method to transform each letter to its uppercase form while preserving any non-letter characters?

Sample input:

String name = “veltech”;

Sample output:
Uppercase word = “VELTECH”

96. You need to write a method that converts all the letters in a given string to lowercase. How will you design your method to ensure every letter is transformed to its lowercase form while leaving any non-letter characters unchanged?

Sample input:

String name = “vel123Tech”;

Sample output:
Transformed word = “VEL123TECH”

Lab Session 36

97. You need to write a method that counts and prints the frequency of each character present in a given string. How will you design your method to tally how many times each character appears and display the results?

Consider the input= “ABCC”.

Expected answer:

A = 1 as char ‘A’ occur 1 time.
B = 1 as char ‘B’ occur 1 time.
C = 2 as char ‘C’ occurs 2 times.

98. Write a method to print characters which occurs maximum number of times in the array. In case of dilemma, print the char which is alphabetically low.

Consider the input as “ABCC”.

The output is ‘C’ as it occurs maximum number of times.

99. Write a method to print the characters which occurs minimum number of times in the array. In case of dilemma, print the char which is alphabetically earlier.

Consider the input as “ABCC”.

The output is ‘A’ as it occurs minimum number of times.

Please note that ‘A’ and ‘B’ occurs same number of times but ‘A’ is earlier than ‘B’.

100. You need to write a method that identifies and prints the unique characters from a given array. How will you design your method to find and display characters that appear only once in the array?

Consider the input as “ABCC”.

The unique characters are ‘A’, ‘B’ and ‘C’.

101. You need to write a method that sorts a given array in ascending order. How will you design your method to arrange the elements from the smallest to the largest value?

Sample input:

[4, 2, 7, 1, 9, 3]

Sample output:

[1, 2, 3, 4, 7, 9]

102. You need to write a method that sorts a given array in descending order. How will you design your method to arrange the elements from the largest to the smallest value?

Sample input: [4, 2, 7, 1, 9, 3]

Sample output: [9, 7, 4, 3, 2, 1]

103. You need to write a method that reverses the order of elements in a given integer array. How will you design your method to flip the array so that the last element becomes the first, and so on?

Sample input:

[4, 2, 7, 1, 9, 3]

Sample output:

[3, 9, 1, 7, 2, 4]

104. You need to write a method that toggles a given string, converting all lowercase letters to uppercase and all uppercase letters to lowercase. How will you design your method to transform each letter to its opposite case while leaving non-letter characters unchanged?

Sample input= “welcoME”

Sample output =”WELCOme”

105. You need to write a method that converts lowercase letters to uppercase and uppercase letters to lowercase in a given string, but leaves vowels unchanged. How will you design your method to make these conversions while preserving the vowels in their original case?

Consider the input as “WELCome”.

The output is: “wElcoMe”. Note that, vowels are maintained without any changes in the output. And, lower case characters are converted into upper case and vice versa.

106. **Imagine you are a detective in a world where all the letters have mysteriously turned into uppercase. Your task is to help restore them to their natural, lowercase state. You are given a collection of characters, each one in uppercase. Your mission is to write a method that will convert all these characters into their lowercase counterparts.**

Here’s the challenge:

You are provided with an array of characters, each in uppercase. Your goal is to create a method that will transform each character in the array to its corresponding lowercase version. After transformation, you should return the new array containing all lowercase characters.

For example:

Input : ['A', 'B', 'C', 'D']

Output : ['a', 'b', 'c',’d’]

Can you solve the case and reveal the lowercase array?

107. You’ve stumbled upon a mysterious string, and you need to determine if it reads the same forward and backward, ignoring case differences. A palindrome is a sequence of characters that reads the same backward as forward.

Your Task:
Write a method to check whether the given string is a palindrome.
The check should be case insensitive, meaning 'A' should be treated as equal to 'a', 'B' as 'b', and so on.

Example:

Input: "Racecar"

Output: True (since "racecar" reads the same forward and backward)

Input: "Hello"
Output: False (since "hello" does not read the same backward)
Can you crack the code and identify whether the string is a palindrome?

Explanation: A word is called as palindrome, if the reverse of the word is same as the original.

```
boolean isPalindrome (String s) {
// write code
}
```

Sample Input/Output-1: isPalindrome("wow") → true
Sample Input/Output-2: isPalindrome("woW") → false
Sample Input/Output-3: isPalindrome("win") →false

Lab Session 39

108. The Mirror Mystery

You've encountered a cryptic message that you suspect might be a palindrome—a sequence that reads the same forwards and backwards. However, this message is a bit tricky, as its case-sensitive, and you need to solve the mystery by treating it case-insensitively.

Your Challenge:
Decrypt the Message: You are given a string, and you need to check if it's a palindrome.

Case Insensitivity: Remember, 'A' should be considered the same as 'a', 'B' as 'b', and so on.

Comparison: The string should read the same forwards and backwards, ignoring the case of the letters.

Example:
Input: "Madam"
Output: True (since "madam" reads the same forward and backward)
Input: "OpenAI"
Output: False (since "openai" does not read the same backward)
Can you solve the Mirror Mystery and confirm whether the given message is a palindrome?

Sample Input/Output-1: isPalindrome("woW") → true

Sample Input/Output-2: isPalindrome("www") → true
Sample Input/Output-3: isPalindrome("win") →false

109. The Number Palindrome Quest

In a land of numerical mysteries, you've discovered an ancient code that may hold a hidden palindrome! A palindrome is a sequence that reads the same forwards and backwards. Your task is to unravel the mystery of this number.

The Challenge:

1. **Decrypt the Number:** You are given a number, and you need to determine if it is a palindrome.
2. **Palindrome Check:** A number is a palindrome if it remains the same when its digits are reversed.

Your Mission:

Write a method to check if the given number is a palindrome.

Example:

- **Input:** 12321
- **Output:** True (since "12321" reads the same forward and backward)
- **Input:** 12345
- **Output:** False (since "12345" does not read the same backward)

Can you solve the Number Palindrome Quest and find out if the number is indeed a palindrome?

110. The Substring Quest

You've stumbled upon a magical string that holds many hidden secrets. Your mission is to uncover all the possible substrings within this string. A substring is any sequence of characters that appears consecutively within the original string.

The Challenge:

1. **Discover the Substrings:** Your task is to write a method that will reveal every possible substring of the given string.
2. **Print Each Substring:** Display each substring, ensuring that no possible combination is left out.

Example:

- **Given String:** "abc"
- **Expected Output:**
  - a
  - ab
  - abc
  - b
  - bc
  - c

Can you crack the code and unveil all the hidden substrings within the string?

Lab Session 40

111. The Reverse Array Mystery

You've come across a string in an ancient manuscript, and you need to uncover its hidden form. Your task is to transform this string into an array of characters and then reveal the characters in reverse order.

The Challenge:

1. **Convert to Characters:** Write a method to convert the given string into an array of characters.
2. **Reverse the Array:** Print the characters in the reverse order from how they appear in the original string.

Example:

- **Given String:** "hello"
- **Expected Output:**
  - o
  - l
  - l
  - e
  - h

Can you solve the Reverse Array Mystery and unveil the string's reversed character array?

112. The Character Array Transformation

You have a collection of individual characters arranged in an array. Your task is to piece them together to form a coherent string.

The Challenge:

1. **Transform the Array:** Write a method to take the given array of characters and combine them into a single string.
2. **Reveal the String:** Output the resulting string formed from the characters in the array.

Example:

- **Given Array:** ['H', 'e', 'l', 'l', 'o']
- **Expected Output:** "Hello"

Can you crack the code and transform the character array into its string form?

Lab Session 41

113. The Comma-Separated Character Transformation

In a world of characters separated by commas, your task is to transform this list into a new format. You need to process a comma-separated string of characters and produce a transformed result based on a specific requirement.

The Challenge:

1. **Read the Input String:** You are given a string of characters separated by commas.
2. **Transform the Characters:** Apply a specific transformation to each character.
3. **Return the Result:** Output the transformed characters as a new string.

Transformation Rules:

1. **Remove Spaces:** Remove any spaces from the characters if present.
2. **Concatenate Characters:** Concatenate all characters into a single string without commas.
3. **Transform (e.g., Convert to Uppercase):** Convert each character to uppercase (or apply another transformation as specified).

Example:

- **Given Input:** "a, b, c, d, e"
  - **Remove Spaces:** "a,b,c,d,e"
  - **Concatenate Characters:** "abcde"
  - **Transform (to Uppercase):** "ABCDE"
  - **Resulting Output:** "ABCDE"

- **Given Input:** " x , y , z "
  - **Remove Spaces:** "x,y,z"
  - **Concatenate Characters:** "xyz"
  - **Transform (to Uppercase):** "XYZ"
  - **Resulting Output:** "XYZ"

114. The Character Extraction Challenge

In the realm of string mysteries, you are tasked with extracting crucial characters from a given string. Your mission is to retrieve three special characters: the first, the middle, and the last.

The Challenge:

1. **First Character:** Identify the very first character of the string.
2. **Middle Character:** Determine the character that lies exactly in the middle of the string.
3. **Last Character:** Find the final character at the end of the string.

```
String getFirstMiddleLast (String s) {
        // write code
}
```

Rules to Follow:

- If the string has an odd length, the middle character is the exact center.
- If the string has an even length, you can choose either of the two central characters or apply your own rule.

Example:

- **Given String:** "abcdefgh"
  - **First Character:** a
  - **Middle Character:** d (if choosing the earlier middle character in even length strings)
  - **Last Character:** h
- **Given String:** "abcdef"
  - **First Character:** a
  - **Middle Character:** c (if choosing the earlier middle character in even length strings)
  - **Last Character:** f

Can you solve the Character Extraction Challenge and unveil the first, middle, and last characters from the string?

Lab Session 42

115. In the realm of strings, you have been tasked with extracting a special trio of characters from the heart of a string. Your mission is to find and reveal the three characters located in the center of the string.

The Challenge:

1. **Locate the Middle Trio:** Write a method to identify and extract the three characters that are centered within the string.
2. **Handle Different Lengths:** If the string has fewer than three characters, you may need to adjust your approach accordingly.

Rules to Follow:

- For strings with an odd length of 5 or more, the middle trio consists of the three characters around the exact center.
- For strings with an even length of 6 or more, you can either choose the three characters around the left center or the right center, or apply your own consistent rule.

Example:

- **Given String:** "abcdefgh"
  - **Middle Trio:** "cde" (3 characters centered around the middle of the string)
- **Given String:** "abcdef"
  - **Middle Trio:** "bcd" (3 characters centered around the middle of the string)
- **Given String:** "abcd"
  - **Middle Trio:** "bcd" (3 characters starting from the second character, if fewer than 5)

```
String getMiddleThree (String s) {
        // write code
}
```

Can you solve the Middle Trio Extraction and reveal the three characters that lie at the center of the string?

116) The Uppercase Count Quest

You've come across a string that holds a mix of characters, and your task is to uncover the number of uppercase letters hidden within

```
int upperCaseCount (String s) {

        // write code
}
```

The Challenge:

1. **Count the Uppercase Letters:** Write a method to scan through the given string and count how many of the characters are uppercase letters.
2. **Reveal the Count:** Return the total number of uppercase letters found.

Rules to Follow:

- An uppercase letter is any character from 'A' to 'Z'.
- Ignore all lowercase letters, digits, and special characters.

Example:

- **Given String:** "Hello World!"
  - **Uppercase Letters:** H and W
  - **Expected Output:** 2
- **Given String:** "Count The CAPS"
  - **Uppercase Letters:** C, T, C, A, P, S
  - **Expected Output:** 6

Can you crack the Uppercase Count Quest and find out how many uppercase letters are in the string?

Lab Session 43

117) The Lowercase Letter Count Challenge

In the land of strings, your task is to uncover how many lowercase letters are hidden within a given string. The string is filled with various characters, but you need to focus solely on those that are lowercase letters.

The Challenge:

1. **Count the Lowercase Letters:** Write a method to explore the given string and determine how many characters are lowercase letters.
2. **Reveal the Count:** Return the total number of lowercase letters discovered.

Rules to Follow:

- A lowercase letter is any character from 'a' to 'z'.
- Ignore all uppercase letters, digits, and special characters.

Example:

- **Given String:** "Hello World!"
  - **Lowercase Letters:** e, l, l, o, o, r, l, d
  - **Expected Output:** 8
- **Given String:** "Count The Letters"
  - **Lowercase Letters:** o, u, n, t, h, e, e, t, t, e, r, s
  - **Expected Output:** 12

Can you solve the Lowercase Letter Count Challenge and determine how many lowercase letters are in the string?

```
int lowerCaseCount (String s) {
        // write code
}
```

118) The Word Count Conundrum

In a realm filled with strings of text, you are tasked with discovering how many words are hidden within a given string. The words are neatly separated by spaces, and your mission is to count them accurately.

The Challenge:

1. **Count the Words:** Write a method to scan through the given string and determine how many distinct words are present.
2. **Handle Spaces:** Words are separated by spaces, and any extra spaces at the beginning, end, or between words should be handled properly.

Rules to Follow:

- Words are defined as sequences of characters separated by spaces.
- Ignore any extra spaces and ensure that you accurately count the number of words.

Example:

- **Given String:** "The quick brown fox jumps over the lazy dog"
  - **Words:** The, quick, brown, fox, jumps, over, the, lazy, dog
  - **Expected Output:** 9
- **Given String:** " Hello world "
  - **Words:** Hello, world
  - **Expected Output:** 2

Can you crack the Word Count Conundrum and determine the total number of words in the string?

```
int wordCount (String sentence) {
        // write code
}
```

Lab Session 44

119) In a land of sentences and phrases, your task is to extract all the words from a given sentence and return them in an organized manner. Words are separated by spaces, and you need to uncover each word and present them in an array.

The Challenge:

1. **Extract the Words:** Write a method to break down the given sentence into individual words.
2. **Return as Array:** Provide the words in the form of a String array.

Rules to Follow:

- Words are sequences of characters separated by spaces.
- Handle any extra spaces properly, ensuring that you accurately extract and return each word.
- The resulting array should contain each word as a separate element.

Example:

- **Given Sentence:** "The quick brown fox jumps over the lazy dog"
  - **Resulting Array:** ["The", "quick", "brown", "fox", "jumps", "over", "the", "lazy", "dog"]
- **Given Sentence:** " Hello world "
  - **Resulting Array:** ["Hello", "world"]

Can you complete the Word Extraction Expedition and return all the words from the sentence as a String array?

```
String [] getAllWords (String sentence) {
        // write code
}
```

120) The Repeated Characters Riddle

You've encountered a string filled with various characters. Your mission is to uncover which characters appear more than once in the string. Reveal these repeated characters in their own collection.

The Challenge:

1. **Identify Repeated Characters:** Write a method to determine which characters appear more than once in the given string.
2. **Return the Repeats:** Provide these repeated characters in a suitable format.

Rules to Follow:

- Characters are considered repeated if they appear more than once in the string.
- Ensure that each repeated character is listed only once in the result.
- The order of repeated characters should be preserved as they first appear in the string.

Example:

- **Given String:** "programming"
  - **Repeated Characters:** ["r", "g", "m"] (since 'r', 'g', and 'm' appear more than once)
- **Given String:** "hello"
  - **Repeated Characters:** ["l"] (since 'l' appears more than once)

Can you solve the Repeated Characters Riddle and uncover all the characters that are repeated in the string?

```
String getDuplicateChars (String string1) {
        // CODE
}
```

Sample Input/output 1:
Input: "AABBBBBC"
Output: "AB"
Here, 'A' and 'B' are repeated. C is not repeated. Hence, 'A' and 'B' are included in output.

Sample Input/output 2:
Input: "RRR"
Output: "R"
Here, character 'R' is the only repeated char. Hence, "R" is included in output.

121) The Whitespace Removal Challenge

In the land of strings, there exists a string filled with unnecessary whitespace that you need to clear away. Your mission is to remove all spaces and return the string in its cleanest form.

The Challenge:

1. **Remove Whitespace:** Write a method to eliminate all whitespace characters from the given string.
2. **Return the Clean String:** Provide the string with all spaces, tabs, and other whitespace characters removed.

Rules to Follow:

- Whitespace characters include spaces, tabs, and newlines.
- The resulting string should be devoid of any whitespace, but all other characters should remain unchanged.

Can you complete the Whitespace Removal Challenge and return the string with all whitespace removed?

```
String truncate (String input1) {
        // write code
}
```

Sample Input/output 1:

input1 = " Welcome to Java "
answer1 = "WelcometoJava"

Sample Input/output 2:

input2 = " Wel come "
answer2 = "Welcome"

122) The Digit Extraction Quest

In the world of mixed characters and numbers, you have a string that contains various types of characters. Your quest is to uncover and extract only the digits from this string and return them as a new string.

The Challenge:

1. **Extract the Digits:** Write a method to sift through the given string and pick out all the digits.
2. **Return the Digits:** Provide the extracted digits concatenated together as a single string.

Rules to Follow:

- Digits are characters from '0' to '9'.
- Ignore all non-digit characters, including letters and special symbols.
- Maintain the order of the digits as they appear in the original string.

```
int filterDigits (String input1) {
    // input is passed in "input1" argument
    // write the logic
    // return answer
}
```

Can you solve the Digit Extraction Quest and extract all the digits from the string, returning them as a single concatenated string?

Sample Input/output 1:

Consider input1="V12t34u5"
Expected answer = 12345.
Explanation: We are extracting the digits alone and returning them as answer.

Sample Input/output 2:

Consider input1="Vtu17788"
Expected answer = 17788
Explanation: We are extracting the digits alone and returning them as answer.

123) The Duplicate Digit Dilemma

In the land of numbers, you have a number that may contain some digits appearing more than once. Your task is to find out how many different digits are duplicated within this number.

The Challenge:

1. **Identify Duplicate Digits:** Write a method to determine which digits in the given number appear more than once.
2. **Count the Duplicates:** Count how many distinct digits are duplicated in the number.

Rules to Follow:

- Digits are considered duplicated if they appear more than once in the number.
- Each duplicated digit should be counted only once.
- Ignore any non-digit characters and focus only on the digits of the number.

Can you solve the Duplicate Digit Dilemma and count the number of distinct digits that are duplicated in the number?

```
int countDuplicateDigits (int input1) {
        // write the logic
        // return answer
}
```

Sample Input/output 1:

Consider input1=12377.
Expected answer = 1.
Explanation: There is one duplicate digit in the given number.

Sample Input/output 2:

Consider input1=433377.
Expected answer = 2.
Explanation: There are 2 duplicate digits in the given number. 3 and 7 are the duplicate digits.

124) The Unique Character Search

In a string filled with characters, you need to uncover a special character that appears only once. Your mission is to identify the first character that is unique within the string.

The Challenge:

1. **Find the Unique Character:** Write a method to determine which character appears only once in the given string.
2. **Identify the First One:** Among all unique characters, find the first one that appears in the string.

Rules to Follow:

- A character is considered unique if it appears exactly once in the string.
- If no character is unique, handle it accordingly (e.g., return a message or a special value).
- The order of characters' matters, so the first unique character should be identified based on its position in the string.

Can you solve the Unique Character Search and find the first unique character in the string?

```
char firstUniqueChar (String str) {
        //CODE
}
```

Sample Input/output 1:

Consider the string string1="RajaRam Mohan Roy";

In the above string, 'j','h','n', 'y' is not repeated. And, 'j' is the first unique char. So, the answer is 'j'.

Sample Input/output 2:

Consider the string str="1234123";

In the above string,'1','2' and '3' are repeated. '4' is not repeated. So, the answer is '4'.

Lab Session 47

125) The Vowel Extraction Expedition

In a sentence filled with various letters, you need to embark on a quest to extract only the vowels. Your mission is to filter out the vowels (both uppercase and lowercase), concatenate them, and return the result as a single string.

The Challenge:

1. **Extract the Vowels:** Write a method to identify and extract all the vowels from the given sentence.
2. **Concatenate the Vowels:** Combine these vowels into a single string, preserving their order as they appear in the original sentence.

Rules to Follow:

- Vowels are defined as a, e, i, o, u and their uppercase counterparts A, E, I, O, U.
- Ignore all non-vowel characters, including consonants, digits, and special symbols.
- Maintain the order of vowels as they appear in the sentence.

Can you complete the Vowel Extraction Expedition and return the concatenated string of vowels from the sentence?

```
String fliterVowels (String input1) {
        // write code
}
```

Sample Input/output 1:

input1= " I am Java "
answer1= "Iaaa"

Explanation: Only the vowels are extracted from each word and we concat all of them and return the answer.

Sample Input/output 2:

input2 = "Wel come "
answer2 = "eoe"

Explanation: There are 2 words in given string. {"Wel", "come"}
Vowels in ("Wel") = "e"
Vowels in ("come") = "oe"

Finally, we concat all the above and return the answer.

126. The Reversed Words Challenge

In a sentence brimming with words, you need to reverse each individual word and then concatenate them to form a new string. Your challenge is to manipulate each word while maintaining their original order.

The Challenge:

1. **Reverse Each Word:** Write a method to reverse every word in the given sentence.
2. **Concatenate the Reversed Words:** Combine these reversed words into a single string, preserving their order in the sentence.

Rules to Follow:

- Words are sequences of characters separated by spaces.
- Reversing a word means the letters are arranged in the opposite order.
- Maintain the order of words as they appear in the sentence.

Can you solve the Reversed Words Challenge and return the concatenated string of reversed words from the sentence?

```
String reverseAndConcat (String input1) {
      // write code
}
```

Sample Input/output 1:

input1 = " I am Java "
answer1 = "ImaavaJ"

Explanation: There are 3 words in given string. {"I", "am", "Java"}
Reverse("I") = “I”;
Reverse("am") = "ma";
Reverse("Java") = "avaJ"
Finally, we concat all the above results and return the answer.
Explanation:
There are 2 words in given string. {"Wel”, “come"}
Reverse ("Wel") = "leW" ; Revese("come") = "emoc"
Finally, we concat all the above results and return the answer.

127. TCS 2023

One programming language has the following keywords that cannot be used as identifiers: Write a program to find if the given word is a keyword or not.

List of keywords = {break, case, continue, default, defer, else, for, func, goto, if, map, range, return, struct, type, var}

Sample Input #1: defer

Output: defer is a keyword

Sample Input #2: While

Sample Output: while is not a keyword

128. The Email Validation Challenge

In the realm of digital communication, you need to verify whether an email address is valid or not. Your task is to check if the given email follows the proper format and return the result accordingly.

The Challenge:

1. **Validate the Email:** Write a method to check if the provided email address conforms to the standard email format.
2. **Return the Result:** Return true if the email is valid, and false if it is invalid.

Rules to Follow:

- A valid email address should follow this basic format: local_part@domain
  - **Local Part:** The part before the @ symbol.
  - **Domain:** The part after the @ symbol, which should include a domain name and a top-level domain (e.g., .com, .org).
- The domain should contain at least one dot (.) after the @ symbol.
- Ensure the local part and domain do not contain spaces or invalid characters.

```
boolean isValidEmail (String s) {
        //write code
}
```

//**refer** the following link, to know the use of regular expressions

//https://howtodoinjava.com/java/regex/java-regex-validate-email-address/

Example:

- **Given Email:** "example@example.com"
  - **Validation:** true (since it follows the correct format)
- **Given Email:** "invalid-email@.com"
  - **Validation:** false (since the domain part is incorrect)
- **Given Email:** "another@domain"
  - **Validation:** false (since the domain part lacks a top-level domain)

129. The URL Validation Quest

In the digital domain, you need to determine whether a given URL is valid. Your mission is to verify if the URL conforms to the standard format and return the appropriate result.

The Challenge:

1. **Validate the URL:** Write a method to check if the provided URL adheres to a valid format.
2. **Return the Result:** Return true if the URL is valid, and false if it is invalid.

Rules to Follow:

- A valid URL typically follows this basic format: scheme://host:port/path?query#fragment
  - **Scheme:** Indicates the protocol (e.g., http, https, ftp).
  - **Host:** The domain name or IP address of the server.
  - **Port:** (Optional) The port number on the server.
  - **Path:** (Optional) The specific path on the server.
  - **Query:** (Optional) Query parameters for the request.
  - **Fragment:** (Optional) A fragment identifier.
-
- Example formats to check:
  - **Valid URL Examples:**
    - http://example.com
    - https://www.example.com:8080/path/to/resource?query=param#fragment
    - ftp://ftp.example.com/file.txt
  - **Invalid URL Examples:**
    - http://example (missing domain)
    -

https://www.example.com:8080/path/to/resource?query=param# (fragment identifier missing after #)
    - ftp://:8080/file.txt (missing host)
- The URL should not contain spaces and should properly include the necessary components if they are specified.

boolean isValidURL (String s) {

```
boolean isValidURL (String s) {
        //write code
}
```

Can you solve the URL Validation Quest and determine if the given URL is valid or invalid?

Lab Session 49

130. ACCENTURE 2023

The Binary number system only uses two digits, 0 and 1 and number system can be called binary string. You are required to implement the following method:

int OperationsBinaryString(String str); The method accepts a string str as its argument. The string str consists of binary digits separated with an alphabet as follows: A denotes AND operation; B denotes OR operation; C denotes XOR Operation. You are required to calculate the result of the string str, scanning the string to right taking one operation at a time, and return the same.

Note: 1) No order of priorities of operations is required. 2) Length of str is odd

3) If str is NULL or None (in case of Python), return -1

Sample Input: str: 1C0C1C1A0B1

Sample Output:1

Explanation: The alphabets in str when expanded becomes “1 XOR 0 XOR 1 XOR 1 AND 0 OR 1”, result of the expression becomes 1, hence 1 is returned.

Command line arguments

131. Write a program to count the number of command line arguments passed.

132. Write a program that gets a number as a command line input and finds the square of that number using **Integer**.**parseInt** (String arg) method.

Lab Session 50

Bitwise operators

(Assume the length of the binary string is 8.)

133. The Bitwise AND Conundrum

In the realm of binary operations, you have two numbers and need to uncover the result of their bitwise AND operation. Your task is to compute this result and present it.

The Challenge:

1. **Read Two Numbers:** Take two integers as input.
2. **Perform Bitwise AND Operation:** Calculate the result of the bitwise AND operation between these two numbers.
3. **Print the Result:** Output the result of the bitwise AND operation.

Rules to Follow:

- The bitwise AND operation compares each bit of the numbers and returns a new number where a bit is set to 1 only if both corresponding bits of the original numbers are 1.
- For example, given numbers 12 (which is 1100 in binary) and 7 (which is 0111 in binary), the result of the bitwise AND operation will be 4 (which is 0100 in binary).

Example:

- **Given Numbers:** 12 and 7
  - **Bitwise AND Operation Result:** 4 (since 1100 & 0111 = 0100)
- **Given Numbers:** 25 and 30
  - **Bitwise AND Operation Result:** 24 (since 11001 & 11110 = 11000)

Can you solve the Bitwise AND Conundrum and find the result of the bitwise AND operation for the given numbers?

Sample input / output:

Let first number = 4 → in binary → 0000 0100
Let second number = 10 → in binary → 0000 1010

0000 0000

So, the result of the operation is 0.

134. The Bitwise OR Challenge

In the realm of binary operations, you have two numbers and need to compute the result of their bitwise OR operation. Your task is to perform this operation and reveal the outcome.

The Challenge:

1. **Read Two Numbers:** Input two integers.
2. **Perform Bitwise OR Operation:** Calculate the result of the bitwise OR operation between these two numbers.
3. **Print the Result:** Output the result of the bitwise OR operation.

Rules to Follow:

- The bitwise OR operation compares each bit of the numbers and returns a new number where a bit is set to 1 if either of the corresponding bits of the original numbers is 1.
- For example, given numbers 12 (which is 1100 in binary) and 7 (which is 0111 in binary), the result of the bitwise OR operation will be 15 (which is 1111 in binary).

Example:

- **Given Numbers:** 12 and 7
  - **Bitwise OR Operation Result:** 15 (since 1100 | 0111 = 1111)
- **Given Numbers:** 25 and 30
  - **Bitwise OR Operation Result:** 31 (since 11001 | 11110 = 11111)

Can you solve the Bitwise OR Challenge and find the result of the bitwise OR operation for the given numbers?

Sample input / output:

Let the first number = 4. 4 in binary → 0000 0100
Let the second number = 10. 10 in binary → 0000 1010

0000 1110

So, the result of the operation is 14.

Lab Session 51

135. The Bitwise XOR Expedition

In the world of binary operations, your mission is to compute the result of the bitwise XOR operation between two given numbers. Your task is to perform this operation and reveal the final outcome.

The Challenge:

1. **Read Two Numbers:** Input two integers.
2. **Perform Bitwise XOR Operation:** Calculate the result of the bitwise XOR operation between these two numbers.
3. **Print the Result:** Output the result of the bitwise XOR operation.

Rules to Follow:

- The bitwise XOR (exclusive OR) operation compares each bit of the numbers and returns a new number where a bit is set to 1 if exactly one of the corresponding bits of the original numbers is 1.
- For example, given numbers 12 (which is 1100 in binary) and 7 (which is 0111 in binary), the result of the bitwise XOR operation will be 9 (which is 1001 in binary).

Example:

- **Given Numbers:** 12 and 7
  - **Bitwise XOR Operation Result:** 9 (since 1100 ^ 0111 = 1001)
- **Given Numbers:** 25 and 30
  - **Bitwise XOR Operation Result:** 19 (since 11001 ^ 11110 = 10011)

Can you solve the Bitwise XOR Expedition and find the result of the bitwise XOR operation for the given numbers?

Sample input / output:

Let the first number = 7. 7 in binary → 0000 0111
Let the second number = 10. 10 in binary → 0000 1010

```
------------------------
  0000 1101
------------------------
```

So, the result of the operation is 13.

136. The Binary Ones Count Challenge

In the land of binary numbers, you have a number that you need to analyze. Your task is to uncover how many 1s are present in its binary representation.

The Challenge:

1. **Read a Number:** Input a single integer.
2. **Convert to Binary:** Determine the binary equivalent of this number.
3. **Count the Ones:** Calculate the number of 1s in the binary representation.
4. **Print the Result:** Output the count of 1s.

Rules to Follow:

- Convert the integer to its binary form, which will be a string of 0s and 1s.
- Count how many 1s are present in this binary string.
- For example, if the given number is 29, its binary equivalent is 11101, which contains 4 ones.

Example:

- **Given Number:** 29
  - **Binary Equivalent:** 11101
  - **Number of 1's:** 4
- **Given Number:** 15
  - **Binary Equivalent:** 1111
  - **Number of 1's:** 4

Can you solve the Binary Ones Count Challenge and determine the number of 1s in the binary representation of the given number?

Lab Session 52

137. The 1's Complement Quest

In the realm of binary numbers, you need to uncover the 1's complement of a given integer. Your task is to compute the complement by flipping all the bits of its binary representation.

The Challenge:

1. **Read a Number:** Input a single integer.
2. **Convert to Binary:** Determine the binary equivalent of this number.
3. **Compute 1's Complement:** Flip all the bits in the binary representation (0 becomes 1 and 1 becomes 0).
4. **Convert Back to Decimal:** Convert the resulting binary string back to its decimal form.
5. **Print the Result:** Output the decimal value of the 1's complement.

Rules to Follow:

- Convert the integer to its binary form. Ensure that the binary string has the same length as the original binary representation.
- Flip all bits (0 to 1 and 1 to 0).
- Convert the resulting binary string back to a decimal integer.
- Note: For simplicity, if the input number is negative, compute the 1's complement for its positive magnitude and then apply the sign if needed.

Example:

- **Given Number:** 5
  - **Binary Equivalent:** 101
  - **1's Complement:** 010 (flipping the bits)
  - **Decimal Result:** 2
- **Given Number:** 9
  - **Binary Equivalent:** 1001
  - **1's Complement:** 0110 (flipping the bits)
  - **Decimal Result:** 6

Can you solve the 1's Complement Quest and determine the 1's complement of the given number?

Sample input / output:

Let the number = 4. 4 in binary → 0000 0100

Output: 1's complement → 1111 1011

138. The Bit Setting Challenge

In the binary world, you are tasked with modifying a number's binary representation. Your mission is to set the n-th bit from the left to 1 and then print the updated number.

The Challenge:

1. **Read a Number:** Input a single integer.
2. **Specify the Bit Position:** Determine which bit (from the left) you need to set to 1. Assume bit positions start at 1 for the leftmost bit.
3. **Convert to Binary:** Find the binary equivalent of the number.
4. **Set the n-th Bit to 1:** Modify the binary representation by setting the n-th bit from the left to 1.
5. **Convert Back to Decimal:** Convert the modified binary string back to a decimal number.
6. **Print the Result:** Output the updated decimal number.

Rules to Follow:

- The n-th bit is counted from the left, starting at 1.
- If n is larger than the number of bits in the binary representation, pad the binary representation with leading 0s as needed.
- After modifying the bit, convert the updated binary back to its decimal form.

Example:

- **Given Number:** 10 (which is 1010 in binary)
- **Set the 3rd Bit from the Left:** Modify the binary representation to 1110
- **Updated Decimal Number:** 14
- **Given Number:** 7 (which is 0111 in binary)
- **Set the 5th Bit from the Left:** Since the binary number is 0111, pad to 00111 and modify to 10111
- **Updated Decimal Number:** 23

Can you solve the Bit Setting Challenge and determine the new number after setting the n-th bit from the left to 1?

139. The Bit Resetting Quest

In the binary world, you have a number and need to modify its binary representation. Your mission is to reset the n-th bit from the left to 0 and then print the updated number.

The Challenge:

1. **Read a Number:** Input a single integer.
2. **Specify the Bit Position:** Determine which bit (from the left) you need to reset to 0. Assume bit positions start at 1 for the leftmost bit.
3. **Convert to Binary:** Find the binary equivalent of the number.
4. **Reset the n-th Bit to 0:** Modify the binary representation by setting the n-th bit from the left to 0.
5. **Convert Back to Decimal:** Convert the modified binary string back to a decimal number.
6. **Print the Result:** Output the updated decimal number.

Rules to Follow:

- The n-th bit is counted from the left, starting at 1.
- If n is larger than the number of bits in the binary representation, pad the binary representation with leading 0s as needed.
- After modifying the bit, convert the updated binary back to its decimal form.

Example:

- **Given Number:** 15 (which is 1111 in binary)
- **Reset the 3rd Bit from the Left:** Modify the binary representation to 1011
- **Updated Decimal Number:** 11
- **Given Number:** 29 (which is 11101 in binary)
- **Reset the 5th Bit from the Left:** Since the binary number is 11101, pad to 011101 and modify to 001101
- **Updated Decimal Number:** 13

Can you solve the Bit Resetting Quest and determine the new number after resetting the n-th bit from the left to 0?