Capstone Project_ASSIGNMENT-01-JAVA

Write a Java program to print all natural numbers from 1 to n using loop. C program to print first n natural numbers using loop

Example

Input

Input upper limit: 10

Output

Natural numbers from 1 to 10: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

NaturalNumbers.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class NaturalNumbers {

	public static void main(String[] args) {

		Scanner scanner = new Scanner(System.in);

		System.out.print("Input upper limit: ");
		int n = scanner.nextInt();

		System.out.print("Natural numbers from 1 to " + n + ": ");
		for (int i = 1; i <= n; i++) {
			System.out.print(i);
			if (i != n) {
				System.out.print(", ");
			}
		}

	}

}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac NaturalNumbers.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java NaturalNumbers
Input upper limit: 10
Natural numbers from 1 to 10: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
C:\Users\Administrator\Documents\Java_Assignments_Day2>
```

Problem Statement

Write a Java program to print all natural numbers in reverse from n to 1 using for loop.

Example

Input

```
Input N: 10
```

Output

```
Natural numbers from 10-1 in reverse:
10, 9, 8, 7, 6, 5, 4, 3, 2, 1
```

ReverseNaturalNumbers.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class ReverseNaturalNumbers {

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

		System.out.print("Input N: ");
		int n = scanner.nextInt();

		System.out.print("Natural numbers from " + n + "-1 in reverse: ");
		for (int i = n; i >= 1; i--) {
			System.out.print(i);
			if (i != 1) {
				System.out.print(", ");
			}
		}

	}

}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac ReverseNaturalNumbers.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java ReverseNaturalNumbers
Input N: 10
Natural numbers from 10-1 in reverse: 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
C:\Users\Administrator\Documents\Java_Assignments_Day2>
```

Problem Statement

Write a Java program to print all even numbers from 1 to n using for loop.

Example

Input

```
Input upper range: 10
```

Output

```
Even numbers between 1 to 10:
2, 4, 6, 8, 10
```

EvenNumbers.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class EvenNumbers {

	public static void main(String[] args) {

		Scanner scanner = new Scanner(System.in);

		System.out.print("Input upper range: ");
		int n = scanner.nextInt();

		System.out.print("Even numbers between 1 to " + n + ":\n");
		for (int i = 2; i <= n; i += 2) {
			System.out.print(i);
			if (i < n - 1) {
				System.out.print(", ");
			}
		}

	}

}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac EvenNumbers.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java EvenNumbers
Input upper range: 10
Even numbers between 1 to 10:
2, 4, 6, 8, 10
C:\Users\Administrator\Documents\Java_Assignments_Day2>
```

Problem Statement

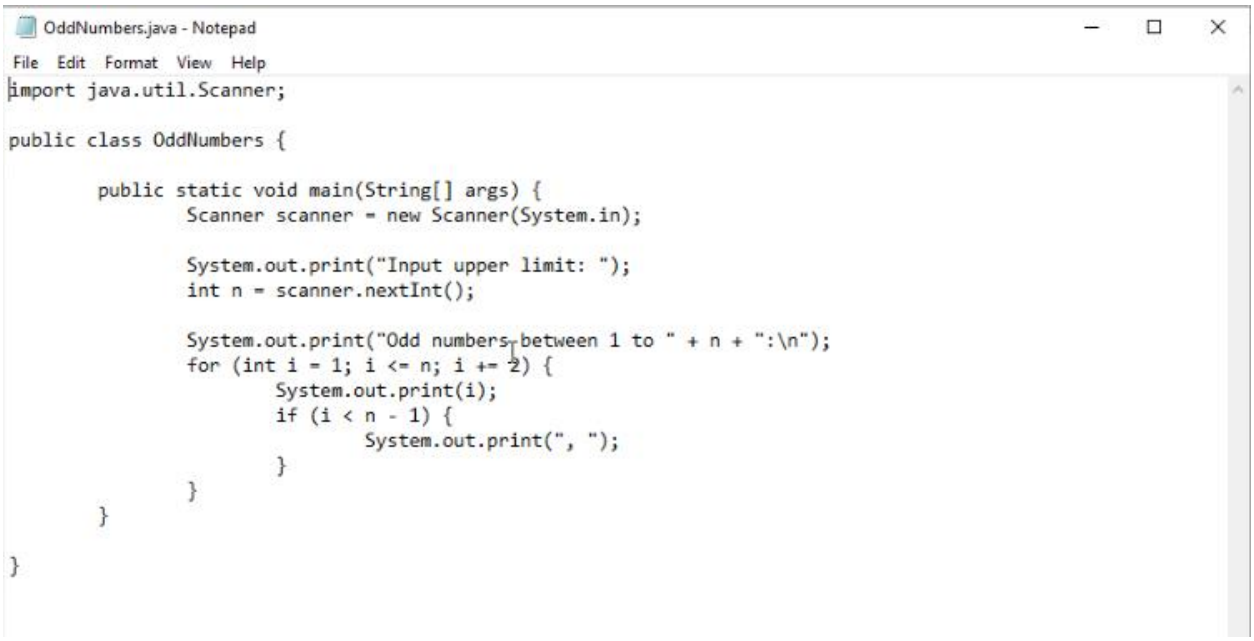
Write a Java program to print all odd numbers from 1 to n using for loop. **Example**

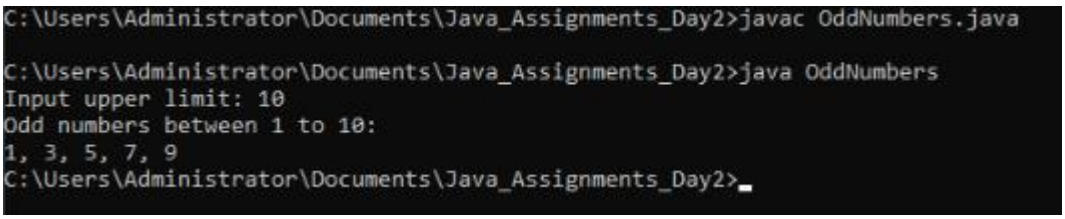
Input

```
Input upper limit: 10
```

Output

```
Odd numbers between 1 to 10:
1, 3, 5, 7, 9
```





Problem Statement

Write a java program to find the sum of all natural numbers between 1 to n using for loop.

Example

Input

```
Input upper limit: 10
```

Output

```
Sum of natural numbers 1-10: 55
```

SumOfNaturalNumbers.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class SumOfNaturalNumbers {

        public static void main(String[] args) {

                Scanner scanner = new Scanner(System.in);

        System.out.print("Input upper limit: ");
        int n = scanner.nextInt();

        int sum = 0;
        for (int i = 1; i <= n; i++) {
            sum += i;
        }

        System.out.println("Sum of natural numbers 1-" + n + ": " + sum);

        }

}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac SumOfNaturalNumbers.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java SumOfNaturalNumbers
Input upper limit: 10
Sum of natural numbers 1-10: 55

C:\Users\Administrator\Documents\Java_Assignments_Day2>
```

Problem Statement

Write a Java program to input number from user and find sum of all even numbers between 1 to n.

Example

Input

```
Input upper limit of even number: 10
```

Output

```
Sum of even numbers between 1 to 10: 30
```

SumOfEvenNumbers.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class SumOfEvenNumbers {

	public static void main(String[] args) {

		Scanner scanner = new Scanner(System.in);

	System.out.print("Input upper limit of even number: ");
	int n = scanner.nextInt();

	int sum = 0;
	for (int i = 2; i <= n; i += 2) {
	    sum += i;
	}

	System.out.println("Sum of even numbers between 1 to " + n + ": " + sum);

	}

}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac SumOfEvenNumbers.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java SumOfEvenNumbers
Input upper limit of even number: 10
Sum of even numbers between 1 to 10: 30
```

Problem Statement

Write a Java program to input a number from user and count number of digits in the given integer using loop.

Example

Input

```
Input num: 35419
```

Output

```
Number of digits: 5
```

CountDigits.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class CountDigits {

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

        System.out.print("Input num: ");
        int num = scanner.nextInt();

        int count = 0;
        int temp = num;

        while (temp != 0) {
            temp = temp / 10;
            count++;
        }

        System.out.println("Number of digits: " + count);

        }

}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac CountDigits.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java CountDigits
Input num: 35419
Number of digits: 5
```

Problem Statement

Write a Java program to input a number from user and find reverse of the given number.

Example

Input

```
Input number: 12345
```

Output

```
Reverse of 12345 = 54321
```

ReverseNumber.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class ReverseNumber {

        public static void main(String[] args) {

                Scanner scanner = new Scanner(System.in);

        System.out.print("Input number: ");
        int number = scanner.nextInt();

        int originalNumber = number;
        int reversedNumber = 0;

        while (originalNumber != 0) {
            int digit = originalNumber % 10;
            reversedNumber = reversedNumber * 10 + digit;
            originalNumber = originalNumber / 10;
        }

        System.out.println("Reverse of " + number + " = " + reversedNumber);

        }

}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac ReverseNumber.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java ReverseNumber
Input number: 12345
Reverse of 12345 = 54321
```

Problem Statement

Write a Java program to find power of a number using for loop. How to find power of a number without using built in library functions

Logic to find power of any number

1. Input base and exponents from user. Store it in two variables say *base* and *expo*.
2. Declare and initialize another variable to store power say power = 1.
3. Run a loop from 1 to *expo*, increment loop counter by 1 in each iteration. The loop structure must look similar to for(i=1; i<=expo; i++).
4. For each iteration inside loop multiply *power* with *num* i.e. power = power * num.
5. Finally after loop you are left with power in *power* variable.

Example

Input

Input base: 2
Input exponent: 5

Output

2 ^ 5 = 32

PowerOfNumber.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class PowerOfNumber {

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

	System.out.print("Input base: ");
	int base = scanner.nextInt();

	System.out.print("Input exponent: ");
	int exponent = scanner.nextInt();

	long power = 1;

	for (int i = 1; i <= exponent; i++) {
	    power *= base;
	}

	System.out.println(base + " ^ " + exponent + " = " + power);

	}

}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac PowerOfNumber.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java PowerOfNumber
Input base: 2
Input exponent: 5
2 ^ 5 = 32
```

Problem Statement

Write a Java program to input a number from user and find all factors of the given number using for loop.

Step by step descriptive logic to find all factors of a number.

1. Input number from user. Store it in some variable say *num*.
2. Run a loop from 1 to *num*, increment 1 in each iteration. The loop structure should look like for(i=1; i<=num; i++).
3. For each iteration inside loop check current counter loop variable *i* is a factor of *num* or not. To check factor we check divisibility of number by performing modulo division i.e. if(num % i == 0) then *i* is a factor of *num*.

   If *i* is a factor of *num* then print the value of *i*.

Example

Input

```
Input number: 12
```

Output

```
Factors of 12: 1, 2, 3, 4, 6, 12
```

FactorsOfNumber.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class FactorsOfNumber {

        public static void main(String[] args) {

                Scanner scanner = new Scanner(System.in);

        System.out.print("Input number: ");
        int num = scanner.nextInt();

        System.out.print("Factors of " + num + ": ");
        for (int i = 1; i <= num; i++) {
            if (num % i == 0) {
                System.out.print(i + ", ");
            }
        }

        }

}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac FactorsOfNumber.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java FactorsOfNumber
Input number: 12
Factors of 12: 1, 2, 3, 4, 6, 12,
```

Problem Statement

Write a Java program to input number from user and check whether number is Strong number or not.

Step by step descriptive logic to check strong number.

- Input a number from user to check for strong number. Store this in a variable say num. Copy it to a temporary variable for calculations purposes, say originalNum = num.
- Initialize another variable to store sum of factorial of digits, say sum = 0.
- Find last digit of the given number num. Store the result in a variable say lastDigit = num % 10.
- Find factorial of lastDigit. Store factorial in a variable say fact.
- Add factorial to sum i.e. sum = sum + fact.
- Remove last digit from num as it is not needed further.
- Repeat steps 3 to 6 till num > 0.
- After loop check condition for strong number. If sum == originalNum, then the given number is Strong number otherwise not.

Example

Input

```
Input number: 145
```

Output

```
145 is STRONG NUMBER
```

StrongNumber.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class StrongNumber {

        public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.print("Input number: ");
        int num = scanner.nextInt();
        int originalNum = num;
        int sum = 0;
        while (num > 0) {
            int digit = num % 10;
            sum += factorial(digit);
            num /= 10;
        }

        if (sum == originalNum) {
            System.out.println(originalNum + " is STRONG NUMBER");
        } else {
            System.out.println(originalNum + " is NOT STRONG NUMBER");
        }
        scanner.close();
    }
    public static int factorial(int n) {
        if (n == 0 || n == 1) {
            return 1;
        } else {
            int fact = 1;
            for (int i = 2; i <= n; i++) {
                fact *= i;
            }
            return fact;
        }

        }
}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac StrongNumber.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java StrongNumber
Input number: 145
145 is STRONG NUMBER
```

Problem Statement

Write a Java program to print Fibonacci series up to n terms using loop.

What is Fibonacci series?

Fibonacci series is a series of numbers where the current number is the sum of previous two terms. For Example: 0, 1, 1, 2, 3, 5, 8, 13, 21, … , (n-1th + n-2th)

Step by step descriptive logic to print n Fibonacci terms.

- Input number of Fibonacci terms to print from user. Store it in a variable say terms.
- Declare and initialize three variables, I call it as Fibonacci magic initialization. a=0, b=1 and c=0.
- Here c is the current term, b is the n-1th term and a is n-2th term.
- Run a loop from 1 to terms, increment loop counter by 1. The loop structure should look like for(i=1; i<=term; i++). It will iterate through n terms
- Inside the loop copy the value of n-1th term to n-2th term i.e. a = b.
- Next, copy the value of nth to n-1th term b = c.
- Finally compute the new term by adding previous two terms i.e. c = a + b.
- Print the value of current Fibonacci term i.e. c

Example

Input

Input number of terms: 10
Output

Fibonacci series:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34

FibonacciSeries.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class FibonacciSeries {

	public static void main(String[] args) {

		Scanner scanner = new Scanner(System.in);

		System.out.print("Input number of terms: ");
		int terms = scanner.nextInt();

		int a = 0;
		int b = 1;
		int c;

		System.out.print("Fibonacci series: ");
		System.out.print(a + ", " + b);

		for (int i = 3; i <= terms; i++) {
			c = a + b;
			System.out.print(", " + c);
			a = b;
			b = c;
		}
		System.out.println();
	}
}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac FibonacciSeries.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java FibonacciSeries
Input number of terms: 10
Fibonacci series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34
```

Problem Statement

Implement a program to calculate the factorial of a given number.

Sample Input and Output

Sample Input	Expected Output
5	120
7	5040

Factorial.java - Notepad

File Edit Format View Help

```
public class Factorial {
        public static void main(String[] args) {
        int[] sampleInputs = {5, 7};

        for (int number : sampleInputs) {
            System.out.println("The factorial of " + number + " is " + factorial(number));
        }
    }

    public static int factorial(int n) {
        if (n == 0) {
            return 1;
        } else {
            int result = 1;
            for (int i = 1; i <= n; i++) {
                result *= i;
            }
            return result;
        }
    }

}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac Factorial.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java Factorial
The factorial of 5 is 120
The factorial of 7 is 5040
```

Problem Statement

Implement a program to display the geometric sequence as given below for a given value n, where n is the number of elements in the sequence.

1, 2, 4, 8, 16, 32, 64, ..., 1024

Sample Input and Output

Sample Input	Expected Output
5	1, 2, 4, 8, 16
8	1, 2, 4, 8, 16, 32, 64, 128

GeometricSequence.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class GeometricSequence {

        public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements in the sequence: ");
        int n = scanner.nextInt();
        scanner.close();
        int current = 1;
        final int maxTerm = 1024;
        final int ratio = 2;

        System.out.print("Geometric sequence: ");
        int count = 0;
        while (current <= maxTerm && count < n) {
            System.out.print(current);

            count++;
            if (count < n) {
                System.out.print(", ");
            }
            current *= ratio;
        }
        System.out.println();
        }
}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac GeometricSequence.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java GeometricSequence
Enter the number of elements in the sequence: 5
Geometric sequence: 1, 2, 4, 8, 16

C:\Users\Administrator\Documents\Java_Assignments_Day2>java GeometricSequence
Enter the number of elements in the sequence: 8
Geometric sequence: 1, 2, 4, 8, 16, 32, 64, 128
```

Problem Statement

Implement a program to check whether a given number is a palindrome.

Palindrome is a sequence that reads the same backwards as forwards.

E.g.: 121, 1331, 2332, 78900987, 123456654321, etc.

Sample Input and Output

Sample Input	Expected Output
1331	1331 is a palindrome
46763	46763 is not a palindrome

Palindrome.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;
public class Palindrome {

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

		System.out.print("Enter a number to check if it is a palindrome: ");
		long number = scanner.nextLong();
		scanner.close();

		String numStr = Long.toString(number);

		int n = numStr.length();

		boolean isPalindrome = true;

		for (int i = 0; i < n / 2; i++) {
			if (numStr.charAt(i) != numStr.charAt(n - 1 - i)) {
				isPalindrome = false;
				break;
			}
		}

		if (isPalindrome) {
			System.out.println(number + " is a palindrome.");
		} else {
			System.out.println(number + " is not a palindrome.");
		}
	}
}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac Palindrome.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java Palindrome
Enter a number to check if it is a palindrome: 1331
1331 is a palindrome.

C:\Users\Administrator\Documents\Java_Assignments_Day2>java Palindrome
Enter a number to check if it is a palindrome: 46763
46763 is not a palindrome.
```

Problem Statement

Implement a program to find out whether a number is divisible by the sum of its digits.

Display appropriate messages.

Sample Input and Output

Sample Input	Expected Output
2250	2250 is divisible by sum of its digits
123	123 is not divisible by sum of its digits

DivisibilityBySumOfDigits.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class DivisibilityBySumOfDigits {

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

	System.out.print("Enter a number : ");
	int number = scanner.nextInt();
	scanner.close();

	int sumOfDigits = 0;
	int originalNumber = number;

	while (number != 0) {
	    int digit = number % 10;
	    sumOfDigits += digit;
	    number /= 10;
	}

	if (sumOfDigits != 0 && originalNumber % sumOfDigits == 0) {
	    System.out.println(originalNumber + " is divisible by the sum of its digits" );
	} else {
	    System.out.println(originalNumber + " is not divisible by the sum of its digits");
	}

	}

}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac DivisibilityBySumOfDigits.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java DivisibilityBySumOfDigits
Enter a number : 2250
2250 is divisible by the sum of its digits

C:\Users\Administrator\Documents\Java_Assignments_Day2>java DivisibilityBySumOfDigits
Enter a number : 123
123 is not divisible by the sum of its digits
```

Problem Statement

Implement a program to find out whether a number is a seed of another number.

A number X is said to be a seed of number Y if multiplying X by its every digit equates to Y.

E.g.: 123 is a seed of 738 as 123*1*2*3 = 738

Sample Input and Output

Sample Input	Expected Output
123, 738	123 is a seed of 738
45, 1000	45 is not a seed of 1000

SeedNumber.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class SeedNumber {

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

        System.out.print("Enter the seed number X: ");
        int seed = scanner.nextInt();
        System.out.print("Enter the number Y to check if it is a seed of X: ");
        int numberY = scanner.nextInt();
        scanner.close();

        int product = seed;
        int originalSeed = seed;
        while (seed != 0) {
            int digit = seed % 10;
            product *= digit;
            seed /= 10;
        }

        if (product == numberY) {
            System.out.println(originalSeed + " is a seed of " + numberY);
        } else {
            System.out.println(originalSeed + " is not a seed of " + numberY);
        }

        }

}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac SeedNumber.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java SeedNumber
Enter the seed number X: 123
Enter the number Y to check if it is a seed of X: 738
123 is a seed of 738

C:\Users\Administrator\Documents\Java_Assignments_Day2>java SeedNumber
Enter the seed number X: 45
Enter the number Y to check if it is a seed of X: 1000
45 is not a seed of 1000
```

Problem Statement

Implement a program to check whether a given number is an Armstrong number.

An Armstrong number is an n-digit number that is equal to the sum of the nth powers of its individual digits.

E.g.: 371 is an Armstrong number as 33 + 73 + 13=371

1634 is an Armstrong number as 14 + 64 + 34+ 44=1634

Hint

Use Math.pow(double a, double b) method to calculate the power of a number

Sample Input and Output

Sample Input	Expected Output
371	371 is an Armstrong number
1635	1635 is not an Armstrong number

ArmstrongNumber.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class ArmstrongNumber {
        public static void main(String[] args) {
                Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number to check if it is an Armstrong number: ");
        int number = scanner.nextInt();
        scanner.close();
        int originalNumber = number;
        int numDigits = countDigits(number);
        int sum = 0;
        while (number > 0) {
            int digit = number % 10;
            sum += Math.pow(digit, numDigits);
            number /= 10;
        }
        if (sum == originalNumber) {
            System.out.println(originalNumber + " is an Armstrong number.");
        } else {
            System.out.println(originalNumber + " is not an Armstrong number.");
        }
    }
    public static int countDigits(int number) {
        int count = 0;
        while (number != 0) {
            number /= 10;
            count++;
        }
        return count;
        }
}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>java ArmstrongNumber
Enter a number to check if it is an Armstrong number: 371
371 is an Armstrong number.

C:\Users\Administrator\Documents\Java_Assignments_Day2>java ArmstrongNumber
Enter a number to check if it is an Armstrong number: 1635
1635 is not an Armstrong number.
```

Problem Statement

Implement a program to check whether a given number is a lucky number.

A lucky number is a number whose sum of squares of every even-positioned digit (starting from the second position) is a multiple of 9.

E.g. - 1623 = 62+32 = 45 is a multiple of 9 and hence is a lucky number.

Sample Input and Output

Sample Input	Expected Output
1623	The number 1623 is a lucky number
15	The number is not a lucky number

LuckyNumber.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;

public class LuckyNumber {

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

        System.out.print("Enter a number to check if it is a lucky number: ");
        long number = scanner.nextLong();
        scanner.close();

        String numStr = Long.toString(number);
        int length = numStr.length();
        long sumOfSquares = 0;

        for (int i = 1; i < length; i += 2) {
            int digit = Character.getNumericValue(numStr.charAt(i));
            sumOfSquares += digit * digit;
        }

        if (sumOfSquares % 9 == 0) {
            System.out.println("The number " + number + " is a lucky number.");
        } else {
            System.out.println("The number " +number + " is not a lucky number.");
        }

        }

}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac LuckyNumber.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java LuckyNumber
Enter a number to check if it is a lucky number: 1623
The number 1623 is a lucky number.

C:\Users\Administrator\Documents\Java_Assignments_Day2>java LuckyNumber
Enter a number to check if it is a lucky number: 15
The number 15 is not a lucky number.
```

Given a list of numbers, write a java function which returns true if one of the first 4 elements in the list is 9. Otherwise it should return false.

The length of the list can be less than 4 also.

Sample Input Expected Output

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

[1, 2, 9] True

[1, 2,3,4] False

Assignment20.java - Notepad

File Edit Format View Help

```
public class Assignment20{
    public static void main(String[] args) {
        int[] arr1 = {1, 2, 9, 3, 4};
        int[] arr2 = {1, 2, 9};
        int[] arr3 = {1, 2, 3, 4};

        System.out.println(contains9(arr1));
        System.out.println(contains9(arr2));
        System.out.println(contains9(arr3));
    }

    public static boolean contains9(int[] nums) {
        // Iterate up to the first 4 elements or the length of the array if it's less than 4
        for (int i = 0; i < Math.min(nums.length, 4); i++) {
            if (nums[i] == 9) {
                return true;
            }
        }
        return false;
    }
}
```

```
C:\Users\Administrator\Documents\Java_Assignments_Day2>javac Assignment20.java

C:\Users\Administrator\Documents\Java_Assignments_Day2>java Assignment20
true
true
false
```