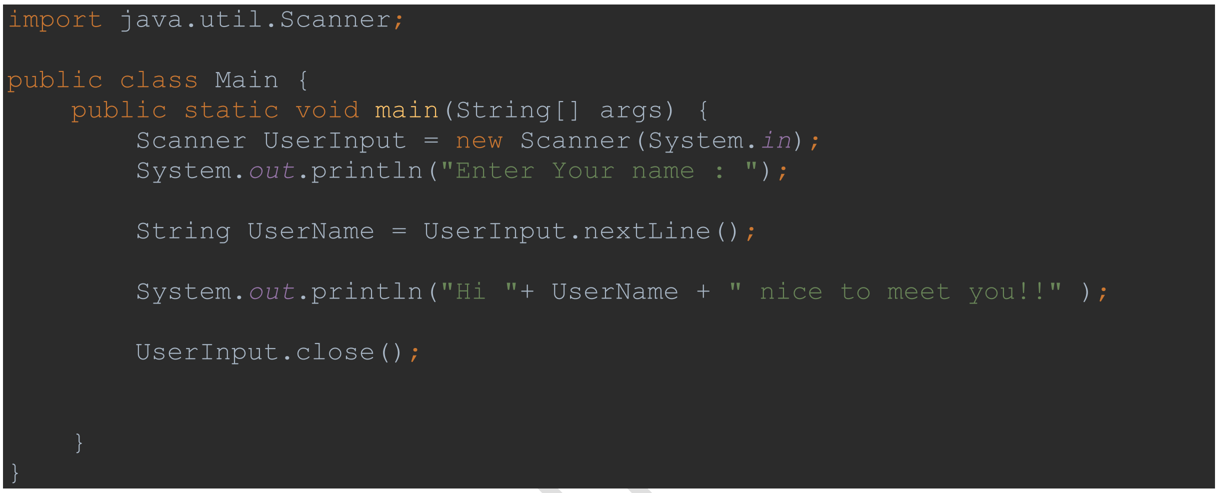
Ex-2. Sequence [E]. Construct pseudocode for a program that asks the user for someone’s name and greets him/her with his/her name. Hint: names can be stored as strings.

```
PROGRAM whatsUpPana
        PRINT “What is your name?”
        GET name
        PRINT “Hi”+ name + “!!”
END
```

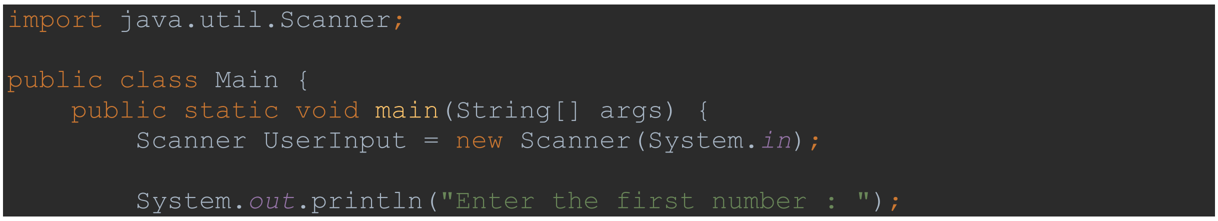


```java
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner UserInput = new Scanner(System.in);
        System.out.println("Enter Your name : ");

        String UserName = UserInput.nextLine();

        System.out.println("Hi "+ UserName + " nice to meet you!!" );

        UserInput.close();


    }
}
```

```
Enter Your name :
Vimukthi
Hi Vimukthi nice to meet you!!
```

Ex-3. Sequence [E]. Write an algorithm in pseudocode that calculates the sum of two numbers and display the result.

```
PROGRAM sum2Values
        READ num1, num2
        COMPUTE sum=num1+num2
        PRINT sum
END
```



```java
import java.util.Scanner;

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

        System.out.println("Enter the first number : ");
        double num1 = UserInput.nextDouble();

        System.out.println("Enter the Second number : ");
        double num2 = UserInput.nextDouble();

        System.out.println("Enter the Third number : ");
        double num3 = UserInput.nextDouble();

        double sum = num1 + num2 + num3;

        System.out.println("Sum of 3 numbers that you've input is "+ sum );
    }
}
```

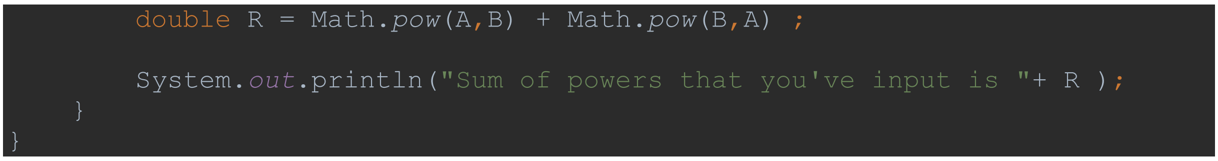
```
Enter the first number :
2.5
Enter the Second number :
12.3
Enter the Third number :
99.85
Sum of 3 numbers that you've input is 114.64999999999999
```

Ex-4. Sequence[E] Write an algorithm in pseudocode that calculates the sum of powers of two numbers A and B as shown by the formula below and display the result.

$R = A^B + B^A$

```
PROGRAM sum2Powers
    READ A, B
    COMPUTE R = A^B + B^A
    PRINT R
END
```

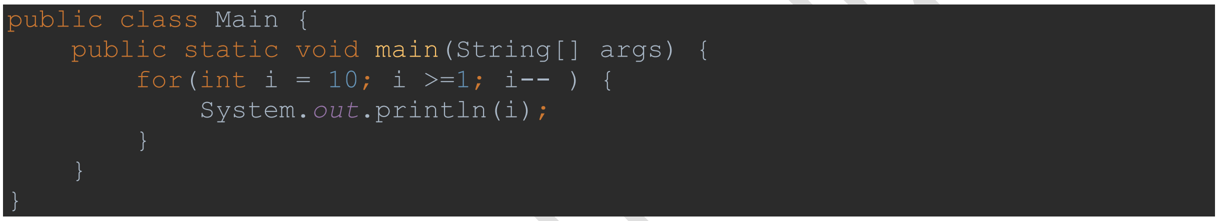
```java
import java.util.Scanner;

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

        System.out.println("Enter the first number : ");
        double A = UserInput.nextDouble();

        System.out.println("Enter the Second number : ");
        double B = UserInput.nextDouble();

```



```
        double R = Math.pow(A,B)  + Math.pow(B,A)  ;

        System.out.println("Sum of powers that you've input is "+ R );
    }
}
```

```
Enter the first number :

2

Enter the Second number :

3

Sum of powers that you've input is 17.0
```

Ex-4. Sequence. Write a flowchart of a program that will display the numbers from 10 backwards to 1.

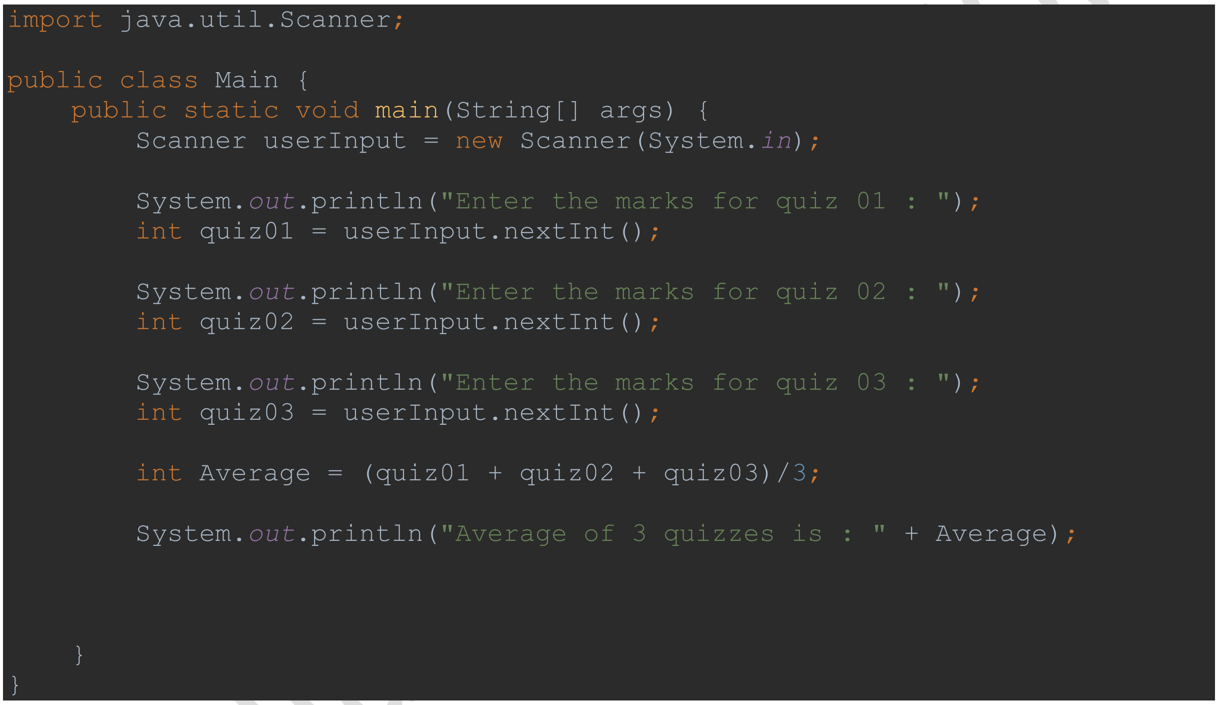


```
public class Main {
    public static void main(String[] args) {
        for(int i = 10; i >=1; i-- ) {
            System.out.println(i);
        }
    }
}
```

```
10

9

8

7

6

5

4

3

2

1
```

Ex-7 Sequence [E]. Write an algorithm in pseudocode that computes the average of five quizzes, and then display the result.

```
PROGRAM quizAverage
        READ quiz1, quiz2, quiz3, quiz4, quiz5
        COMPUTE average = (quiz1+quiz2+quiz3+quiz4+quiz5)/5
        PRINT average
END
```



```java
import java.util.Scanner;

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

        System.out.println("Enter the marks for quiz 01 : ");
        int quiz01 = userInput.nextInt();

        System.out.println("Enter the marks for quiz 02 : ");
        int quiz02 = userInput.nextInt();

        System.out.println("Enter the marks for quiz 03 : ");
        int quiz03 = userInput.nextInt();

        int Average = (quiz01 + quiz02 + quiz03)/3;

        System.out.println("Average of 3 quizzes is : " + Average);



    }
}
```

```
Enter the marks for quiz 01 :
20
Enter the marks for quiz 02 :
20
Enter the marks for quiz 03 :
20
Average of 3 quizzes is : 20
```

Ex-11 Sequence. [E] Write an algorithm in pseudocode that converts Celsius degrees into its equivalent Fahrenheit degrees. Use the formula: F= (9/5) * C+32.

```
Algorithm CelsiusToFahrenheit

Input: Celsius temperature (C)

Output: Fahrenheit temperature (F)


1. Read C  // Input the Celsius temperature

2. F = (9/5) * C + 32  // Use the conversion formula

3. Display F  // Output the equivalent Fahrenheit temperature
```

```java
import java.util.Scanner;

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

        System.out.println("Enter the temperature in Celsius: ");

        double Celsius = scanner.nextDouble();

        double Fahrenheit = (9.0/5) * Celsius + 32;

        System.out.println("Equivalent temperature in Fahrenheit: " +
Fahrenheit);

        scanner.close();
    }
}
```

```
Enter the temperature in Celsius:

18

Equivalent temperature in Fahrenheit: 64.4
```

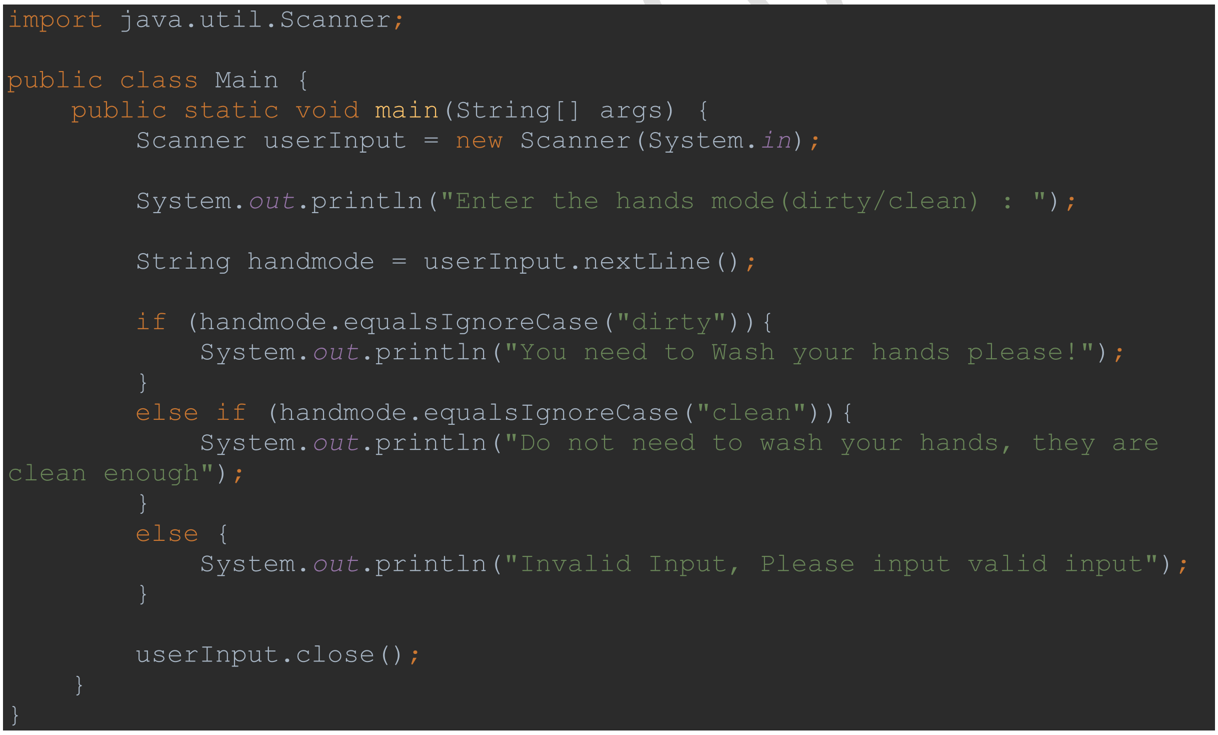
SELECTION

Develop an algorithm for the following problems; use top down design, cursory sketches, flowcharts and pseudocodes for each problem of the following.

In the following answers, we skip the { PROGRAM / END } requirement to make solutions shorter.

Ex-1. Write pseudocode to decide whether or not to wash your hands.

```
INPUT HandsMode % Modes are ‘dirty’ OR ‘clean’

IF HandsMode == ‘dirty’ THEN

        PRINT ‘ Wash your dirty Hands, Please ’

ELSE

        PRINT ‘ Do not need to wash your hands, they are clean enough ’

END
```



```
import java.util.Scanner;

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

        System.out.println("Enter the hands mode(dirty/clean) : ");

        String handmode = userInput.nextLine();

        if (handmode.equalsIgnoreCase("dirty")){
            System.out.println("You need to Wash your hands please!");
        }
        else if (handmode.equalsIgnoreCase("clean")){
            System.out.println("Do not need to wash your hands, they are
clean enough");
        }
        else {
            System.out.println("Invalid Input, Please input valid input");
        }

        userInput.close();
    }
}
```

```
Enter the hands mode(dirty/clean) :

dirtY

You need to Wash your hands please!
```

Ex- 2. Write pseudocode to decide whether or not is time to make lunch. (Please see algorithm in problem #7 and shorten it for this problem)

```
Algorithm CelsiusToFahrenheit

Input: Celsius temperature (C)

Output: Fahrenheit temperature (F)


1. Read C  // Input the Celsius temperature
2. $F = (9/5) * C + 32$  // Use the conversion formula
3. Display F  // Output the equivalent Fahrenheit temperature
```

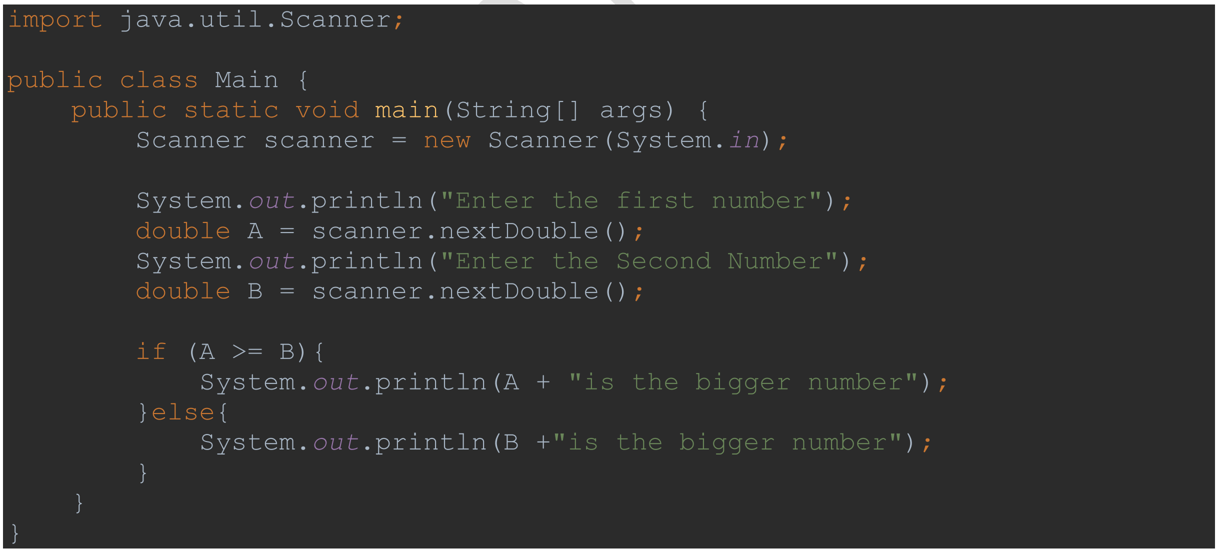
```java
import java.time.LocalTime;

public class LunchDecision {

   // Method to check if it's lunchtime
   private static boolean isLunchTime(LocalTime currentTime, LocalTime startTime, LocalTime endTime) {
      return currentTime.isAfter(startTime) && currentTime.isBefore(endTime);
   }

   public static void main(String[] args) {
      // Get the current time
      LocalTime currentTime = LocalTime.now();

      // Define lunch time range (e.g., from 12:00 PM to 1:00 PM)
      LocalTime lunchStartTime = LocalTime.of(12, 0);
      LocalTime lunchEndTime = LocalTime.of(13, 0);

      // Check if it's lunchtime
      if (isLunchTime(currentTime, lunchStartTime, lunchEndTime)) {
         System.out.println("It's time to make lunch!");
      } else {
            System.out.println("It's not yet time to make lunch. Keep working!");
        }
    }
}
```

Answer:

It's not yet time to make lunch. Keep working!

Write pseudocode to print the largest of two given numbers.

```
INPUT two numbers: A and B
IF A >= B          % notice the symbol >=
    PRINT ' A is largest'
ELSE
    PRINT ' B is largest'
ENDIF
```

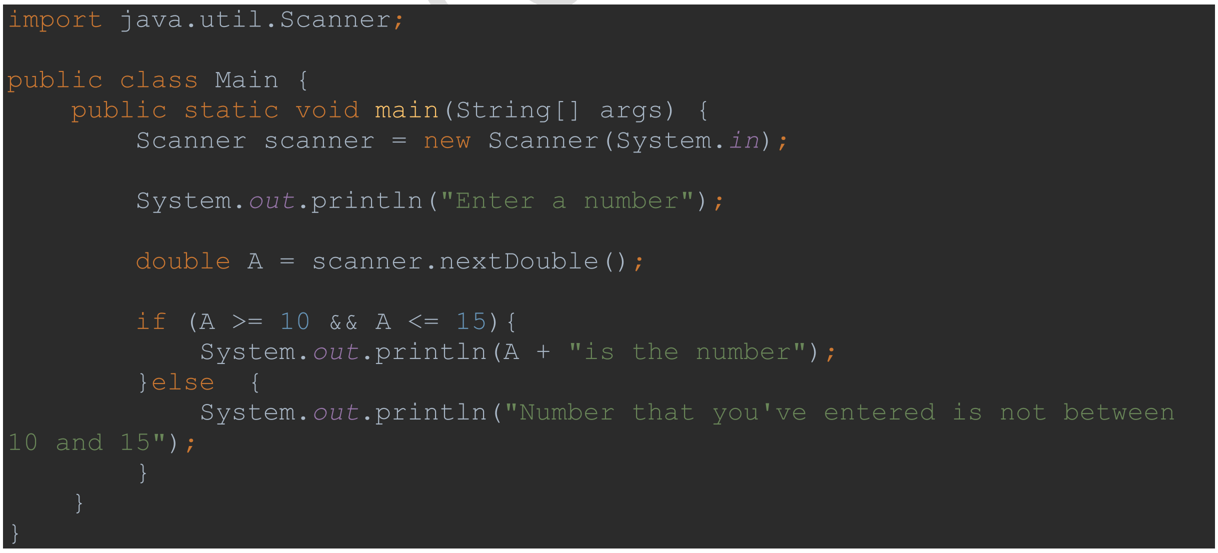


```
import java.util.Scanner;

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

        System.out.println("Enter the first number");
        double A = scanner.nextDouble();
        System.out.println("Enter the Second Number");
        double B = scanner.nextDouble();

        if (A >= B){
            System.out.println(A + "is the bigger number");
        }else{
            System.out.println(B +"is the bigger number");
        }
    }
}
```

```
Enter the first number
10
Enter the Second Number
15
15.0is the bigger number
```

Ex-4. Write an algorithm in pseudocode which ask the user for a number, then decide if the number is between 10 and 15, if it is, print the number.

```
Algorithm CheckNumberRange

Input: None

Output: None


1. Display "Please enter a number:"

2. Read userNumber  // Input the number from the user


3. // Check if the number is between 10 and 15

4. If userNumber is greater than or equal to 10 and userNumber is less than or equal to 15:

5.     Display "The number is: " + userNumber  // Print the number

6. Else:

7.     Display "The number is not between 10 and 15."


8. End Algorithm
```



```
import java.util.Scanner;

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

        System.out.println("Enter a number");

        double A = scanner.nextDouble();

        if (A >= 10 && A <= 15){
            System.out.println(A + "is the number");
        }else  {
            System.out.println("Number that you've entered is not between
10 and 15");
        }
    }
}
```

```
Enter a number

7

Number that you've entered is not between 10 an
```

Ex-6. Selection. Write an algorithm in pseudocode to decide whether or not to take a shower. Then, decide if you shampoo your hair. Shower are taken from 7:30-9:30 am and shampooing only Monday, Wednesday, and Fridays.

```
INPUT time, day
IF time >= 7:30 AND time <=9:30
    IF day is Monday or Wed or Friday
        PRINT 'take a shower & shampoo your hair'
    ELSE
        PRINT 'take a shower skip shampooing your hair'
    END IF
ELSE
    PRINT 'can't take a shower neither shampoo your hair'
ENDIF
```

NOTE: You can only shampoo your hair only if you take a shower and the day is Monday, Wednesday or Friday

```
import java.time.DayOfWeek;
import java.time.LocalDate;
import java.util.Date;

public class Main {
    public static void main(String[] args) {
        Date date = new Date();
        LocalDate localDate = LocalDate.now(); // Get the current date
        LocalDate localDate1 =
date.toInstant().atZone(java.time.ZoneId.systemDefault()).toLocalDate();

//          System.out.println(date.toString());
//          System.out.println(localDate.toString());
        System.out.println(localDate1.toString());

        // Check if the day of the week is Monday, Wednesday, or Friday
        if (isDesiredDay(localDate)) {
            System.out.println(localDate + " is Monday, Wednesday, or
Friday.");
        } else {
            System.out.println(localDate + " is not Monday, Wednesday, or
Friday.");
        }
    }

    // Method to check if the day of the week is Monday, Wednesday, or
Friday
```

```
    private static boolean isDesiredDay(LocalDate date) {
        DayOfWeek dayOfWeek = date.getDayOfWeek();
        return dayOfWeek == DayOfWeek.MONDAY || dayOfWeek ==
DayOfWeek.WEDNESDAY || dayOfWeek == DayOfWeek.FRIDAY;
    }
}
```

2024-01-11

2024-01-11 is not Monday, Wednesday, or Friday.

LOOPS

Design algorithms to solve the following problems. Use designing tools, such as, problem iteration decomposition, trace tables, flowcharts and pseudocodes.

Reading an integer between 1 and 10 from the user with the help of While loop.

```
import java.util.Scanner;

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

        System.out.println("Enter number between 1 and 10 : ");

        int i = scanner.nextInt();

        while ( i <1 || i > 10){
            System.out.println(i + " is not between 1 and 10, Try Again!");
            i = scanner.nextInt(); // you are reading new number form user
        }
        System.out.println( i+ " is between 1 and 10!");

    }
}
```

```
Enter number between 1 and 10 :

-5

-5 is not between 1 and 10, Try Again!

-2

-2 is not between 1 and 10, Try Again!

2

2 is between 1 and 10!
```

```
public class Main {
    public static void main(String[] args) {
        for (int i=1; i <= 5; i++){
            for(int j = 1; j <=i; j++){
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

```
*

**

***

****
```

*****

```
public class Main {
    public static void main(String[] args) {
        for (int i=1; i <= 10; i++){
            for(int j = 1; j <= 10; j++)
                System.out.print((i * j) + " ");

            System.out.println();
        }
    }
}
```

```
1 2 3 4 5 6 7 8 9 10

2 4 6 8 10 12 14 16 18 20

3 6 9 12 15 18 21 24 27 30

4 8 12 16 20 24 28 32 36 40

5 10 15 20 25 30 35 40 45 50

6 12 18 24 30 36 42 48 54 60

7 14 21 28 35 42 49 56 63 70

8 16 24 32 40 48 56 64 72 80

9 18 27 36 45 54 63 72 81 90

10 20 30 40 50 60 70 80 90 100
```

Java continue statement is used to skip the current iteration of a loop. Continue statement in java can be used with for , while and do-while loop.

```
public class Main {
    public static void main(String[] args) {
        for (int i=1; i <= 10; i++){
            if (i % 2 == 0)
                continue;
            System.out.println(i + " ");
        }
    }
}
```

```
1

3

5

7

9
```

Ex-1. Loops. Construct a program to print a message (e.g., Hello There World) of your choice N times.

```
import java.util.Scanner;

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

        System.out.println("Enter number of Greetings: ");
        int counter = scanner.nextInt();

        for (int i = 0; i < counter; i++ ){
            System.out.println((i+1) + " Good Morning Sunshine!");
        }
    }
}
```

Enter number of Greetings:

1

1 Good Morning Sunshine!

Can enhance your program to make it more user-friendly and enjoyable.

Handle Edge Cases: Consider handling edge cases where the user enters a non-positive number or a non-numeric input. You can use a loop to prompt the user until a valid input is provided.

To handle edge cases for invalid input more gracefully, you can catch the **'InputMismatchException'** that may occur if the user enters non-numeric input. Additionally, you can check for a non-positive number separately.

Add Delays Between Greetings: To simulate a more natural interaction, you can add a small delay between each greeting. You can use **Thread.sleep** for this (don't forget to handle exceptions).

In your specific program, the **Thread.sleep(500)** is used to add a small delay between greetings. If, for some reason, the thread is interrupted during this sleep, the catch block will be executed, and a message will be printed, along with the stack trace.

```
import java.util.InputMismatchException;
import java.util.Scanner;

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

        System.out.println("Hello! How many times would you like to hear a
morning greeting?");

        int counter = 0;
        boolean validInput = false;

        while (!validInput) {
            try {
                System.out.print("Enter a positive number: ");
                counter = scanner.nextInt();
                if (counter > 0) {
                    validInput = true;
                } else {
                    System.out.println("Please enter a positive number.");
                }
            } catch (InputMismatchException e) {
                System.out.println("Invalid input. Please enter a valid 
positive number.");
                // Clear the buffer
                scanner.nextLine();
            }
        }

        for (int i = 0; i < counter; i++) {
            System.out.println((i + 1) + " 🌞 Good Morning Sunshine!");
            try {
                // Add a small delay between greetings (500 milliseconds)
                Thread.sleep(500);
            } catch (InterruptedException e) {
                System.out.println("Thread was interrupted while 
sleeping!");
                e.printStackTrace();
            }
        }

        System.out.println("Thank you for spreading positivity with morning 
greetings!");

        // Close the Scanner
        scanner.close();
    }
}
```

```
Hello! How many times would you like to hear a morning greeting?

Enter a positive number: a

Invalid input. Please enter a valid positive number.

Enter a positive number: -1

Please enter a positive number.

Enter a positive number: 0354.5

Invalid input. Please enter a valid positive number.

Enter a positive number: 2

1 🌞 Good Morning Sunshine!

2 🌞 Good Morning Sunshine!

Thank you for spreading positivity with morning greetings!
```

Generate a random target page number for the user to guess, you can use the Random class in Java.

```
import java.util.Scanner;
import java.util.Random;

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

        // Generate a random target page number between 1 and 100
        int targetPage = random.nextInt(100) + 1;

        // Print initial prompt
        System.out.println("Can you guess the target page number between 1
and 100?");

        // Initialize counter
        int counter = 0;

        // Loop until the correct page number is guessed
        while (true) {
            // Input: Ask for a guess
            System.out.print("Enter your guess for the page number: ");
            int guess = scanner.nextInt();

            // Check if the guess is correct
            if (guess == targetPage) {
                System.out.println("You got it!");
                break; // Exit the loop if the guess is correct
            } else {
                // Provide feedback and prompt for another guess
                System.out.println("Try again!");
                if (guess > targetPage) {
                    System.out.println("Too high");
                } else {
                    System.out.println("Too low");
                }
                counter++;
            }
        }

        // Print the number of trials
        System.out.println("It took you " + counter + " trials to find the
correct page number.");

        // Close the Scanner
        scanner.close();
    }
}
```

```
Can you guess the target page number between 1 and 100?
Enter your guess for the page number: 50
Try again!
Too low
Enter your guess for the page number: 75
You got it!
It took you 1 trials to find the correct page number.
```

Write pseudocode for a program which figures out whether a given year is a leap year. In the Gregorian calendar three criteria must be taken into account to identify leap years:

1. The year can be evenly divided by 4, and

2. If the year can be evenly divided by 100, it is NOT a leap year, unless;

3. The year is also evenly divisible by 400. Then it is a leap year.

In general terms, the algorithm for calculating a leap year is as follows... A year will be a leap year if it is divisible by 4 but not by 100. If a year is divisible by 4 and by 100, it is not a leap year unless it is also divisible by 400. this is enough to check if a year is a leap year.

The main difference between while and do-while loop is the order of condition checking and statement execution. While loop checks the condition before executing the statement(s**), do-while loop executes the statement(s) at least once before checking the condition.**

```
import java.util.Scanner;

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

        // Input: Ask the user to enter a year
        int userYear;

        do {
            System.out.print("Enter a year (or enter -1 to stop): ");
            userYear = scanner.nextInt();

            // Check if the user wants to stop
            if (userYear == -1) {
                System.out.println("Program stopped.");
            } else {
                // Check if it's a leap year
                if ((userYear % 4 == 0) && ((userYear % 100 != 0) ||
(userYear % 400 == 0))) {
                    System.out.println(userYear + " is a leap year.");
                } else {
                    System.out.println(userYear + " is not a leap year.");
                }
            }

        } while (userYear != -1);

        // Close the Scanner
        scanner.close();
    }
}
```

```
Enter a year (or enter -1 to stop): 2013

2013 is not a leap year.

Enter a year (or enter -1 to stop): 2016

2016 is a leap year.
```