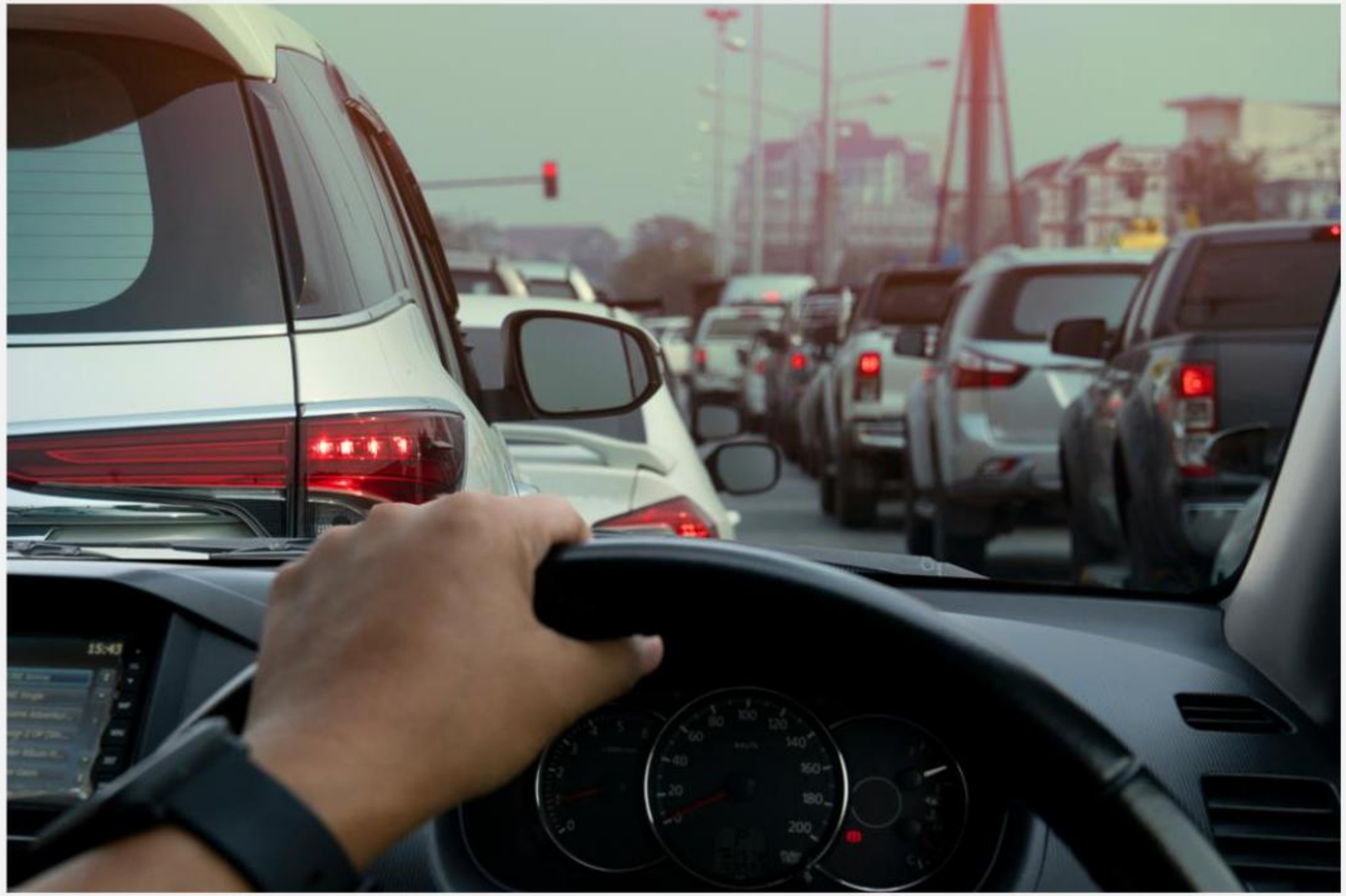
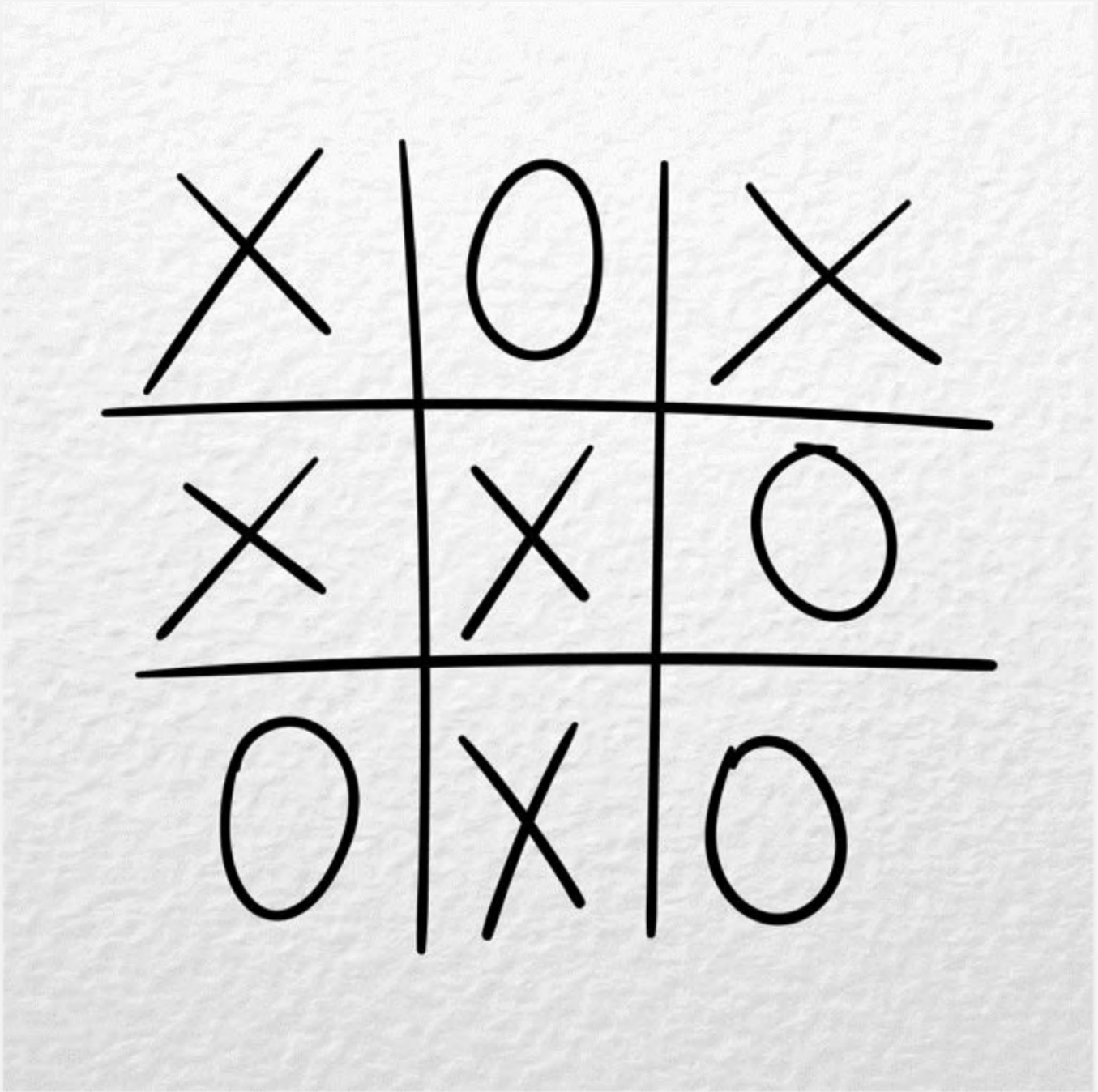


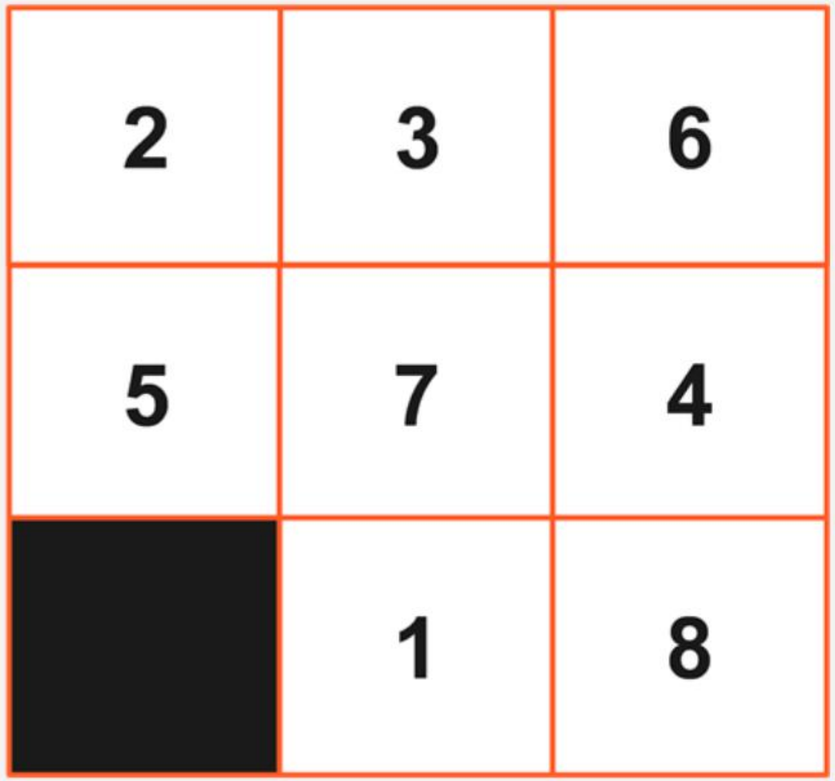
Think and Tell

- What do you do when you get stuck in a traffic jam and have a flight to catch?
- How will you solve this problem if you travel in a hired taxi?

Tic Tac Toe

- Have you ever played a tic-tac-toe game before?
- How do you play this game?
- What strategy should you use to win the game?





Sliding Puzzle

- Have you ever solved a slider puzzle?
- What makes a sliding puzzle so interesting?
- Can a computer solve a 3 × 3 slider puzzle on its own?
- Read more about combination puzzles here.

Learning Objectives

- Break down a complex problem into simple steps
- Use an algorithmic approach to solve problems
- Create pseudocode to represent algorithms
- Analyze the input and output requirements of a computer problem

Learning Objectives

- Break down a complex problem into simple steps
- Use an algorithmic approach to solve problems
- Create pseudocode to represent algorithms
- Analyze the input and output requirements of a computer problem

Learning Objectives

- Break down a complex problem into simple steps
- Use an algorithmic approach to solve problems
- Create pseudocode to represent algorithms
- Analyze the input and output requirements of a computer problem

Learning Objectives

- Break down a complex problem into simple steps
- Use an algorithmic approach to solve problems
- Create pseudocode to represent algorithms
- Analyze the input and output requirements of a computer problem

Learning Objectives

- Break down a complex problem into simple steps
- Use an algorithmic approach to solve problems
- Create pseudocode to represent algorithms
- Analyze the input and output requirements of a computer problem

Algorithmic Problem Solving

What Do These Images Have in Common?

Robotic Pool Cleaner

Robotic Vacuum Cleaner

Drone to Spray Pesticide

To complete the task, all three devices should systematically traverse the entire area efficiently.

Let's proceed with the robotic swimming pool cleaner example.

Cleaning a Swimming Pool

- Each tile should be cleaned until all the tiles are completed inside the swimming pool.
- There may be multiple ways to clean the swimming pool.
- Step by step instructions need to be provided to complete the task.

First Scenario: Instruction Given to a Professional Cleaner

Second Scenario: Instructions Given to an Amateur Cleaner

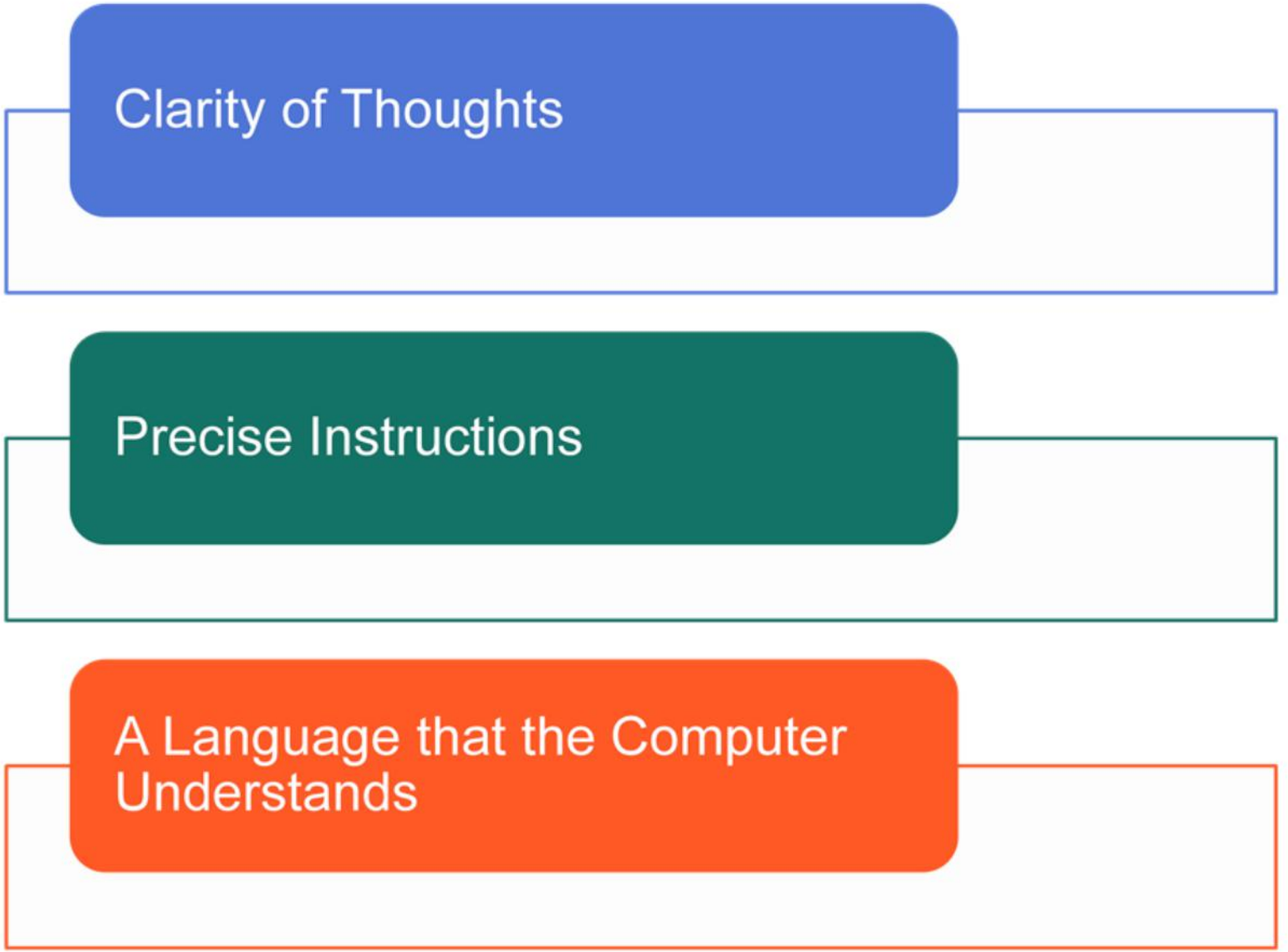
Think and Tell

- Giving instructions to human beings is quite easy as they can understand the common language used for communication.
- Is it possible for a computer to understand the same English language commands?
- The answer is No. Computers cannot think like human beings and decide what to do in each situation.

Think and Tell

- Computers do not have intelligence on their own and therefore cannot make decisions on their own.
- How can a computer be programmed to do a required task?
- If we specify the solution in a language that a computer can understand, will it solve the problem?

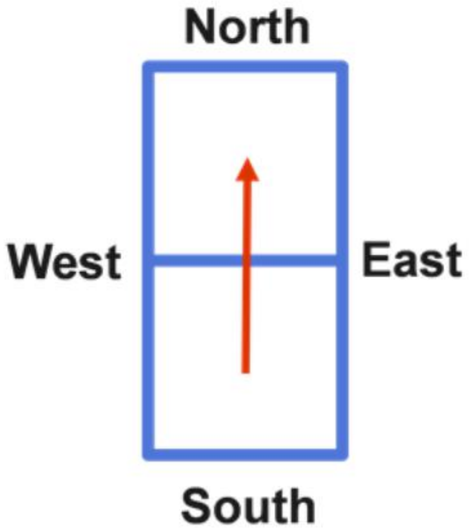
How Can a Computer be Programmed to Solve a Problem?



Consider a Few Scenarios to Begin

Consider there is a single tile, then the steps to clean are:

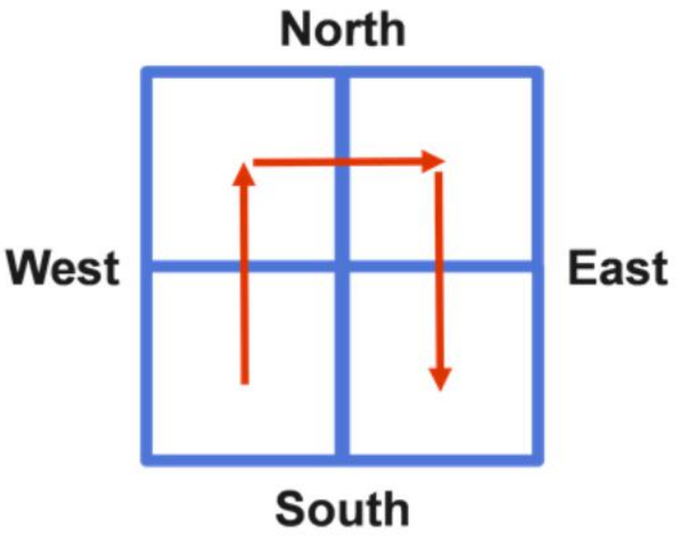
1. Start
2. Clean
3. Stop



Consider there are two tiles, then the steps to clean are:

1. Start
2. Clean
3. Set direction to North
4. Move one tile ahead
5. Clean
6. Stop

Consider Another Scenario to Clean



Consider there are four tiles, then the steps to clean are:

1. Start
2. Clean
3. Set direction to North
4. Move one tile ahead
5. Clean
6. Set direction to east
7. Move one tile ahead
8. Clean
9. Set direction to south
10. Move one tile ahead
11. Clean
12. Stop

Can you identify the pattern where the instructions are being repeated?

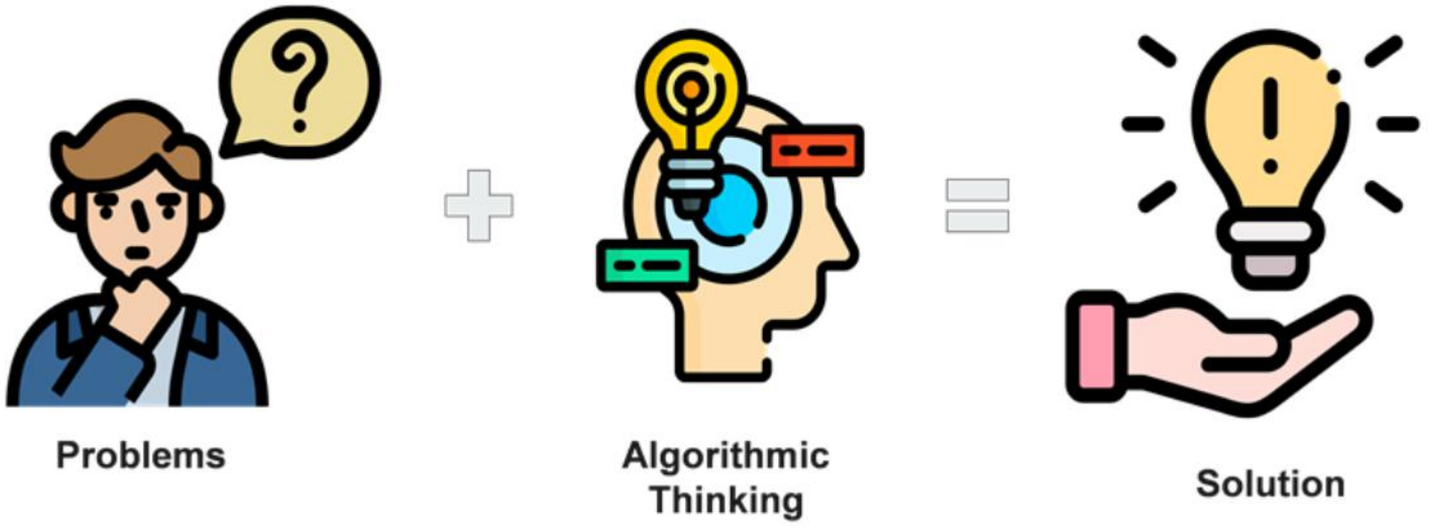
Think and Tell

- Suppose there are 20*30 tiles in the swimming pool, how many instructions should we give to clean the entire pool?
- Start and stop are 2 steps.
- Following are the repeated steps:
  - Clean is repeated for **n** times
  - Move is repeated for **n-1** times
  - Set direction is repeated for **n-1** times
- Therefore, (n)+(n-1)+(n-1)+2 = 3n steps are required.
- Totally, 3*1200 = 3600 instructions must be provided to clean the entire pool.

Can the number of instructions be reduced since there are repetitive steps involved?

Problem: How Can Repetitive Instructions Be Avoided by Identifying Patterns?

- Giving instructions using the Brute Force approach requires every simple step to be mentioned in order to solve the problem. This is not the optimal way.
- Algorithmic thinking or the process of breaking a problem down into repeatable steps is required to solve such problems.



Algorithmic Thinking

- How can you solve a given problem?
  - Define the problem
  - Think of possible solutions
  - Evaluate and select the optimal solution
  - Implement the solution

- Why is problem solving important for a person?
  - Gives the ability to think outside of the box
  - Helps to sharpen logical thinking
  - Lays a strong foundation in programming skills

Problem-Solving Techniques

Algorithms

Pseudocode

Algorithm

What Is an Algorithm?

- It is a set of instructions designed to perform a specific task.
- It provides a simple description of how the problem needs to be solved.
- It lists baby steps towards writing programs in programming languages.
- It is a simple set of English statements.

An algorithm can be handwritten on a piece of paper or on a text editor using a computer.

Let's revisit the problem of cleaning the swimming pool that has 1200 tiles.

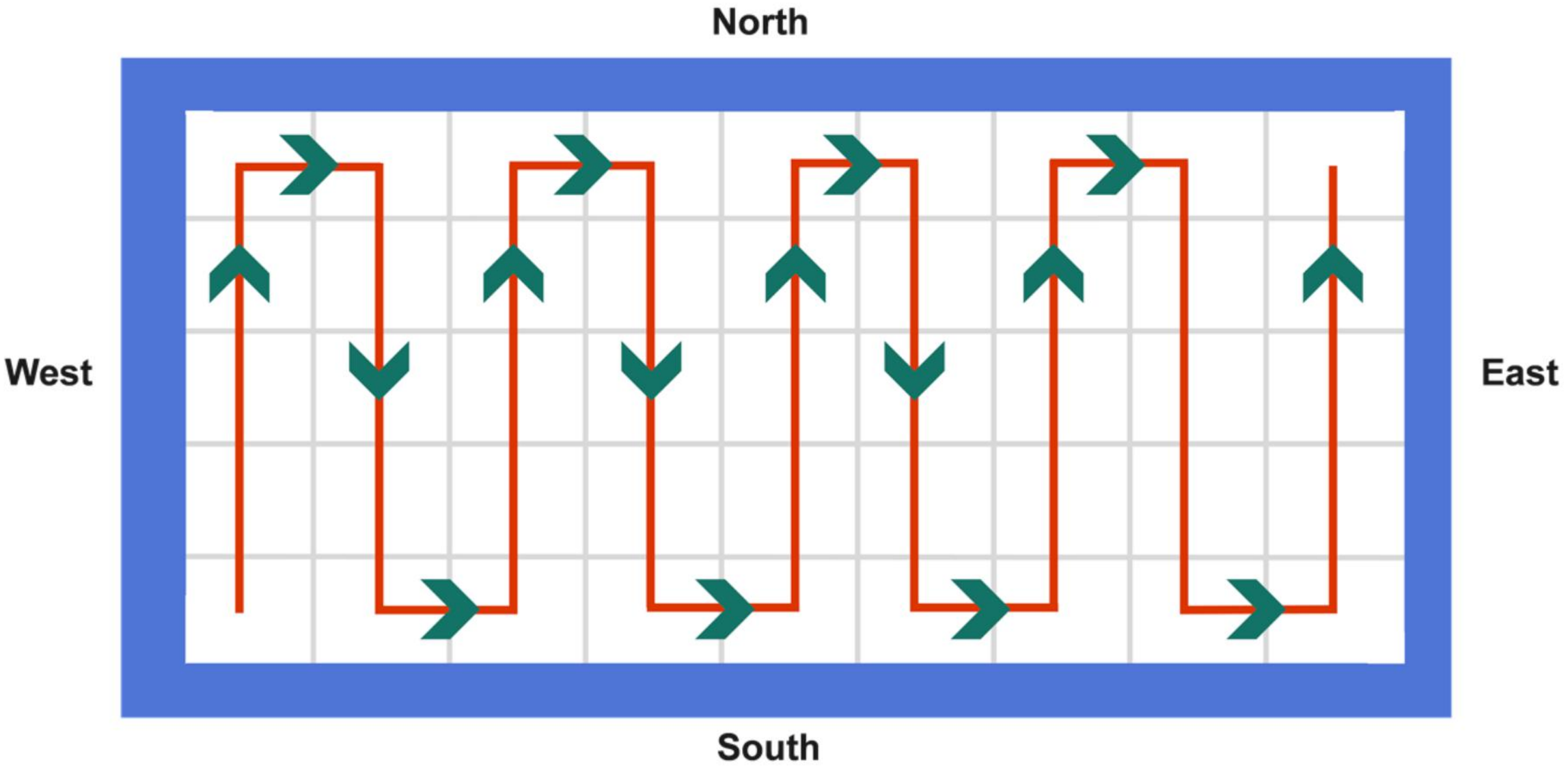
Algorithm to Clean the Pool

Find the solution to clean the pool.

Write an algorithm by listing step-by-step solutions for the given problem.

DEMO

Clean the Pool Starting From the South-West Corner



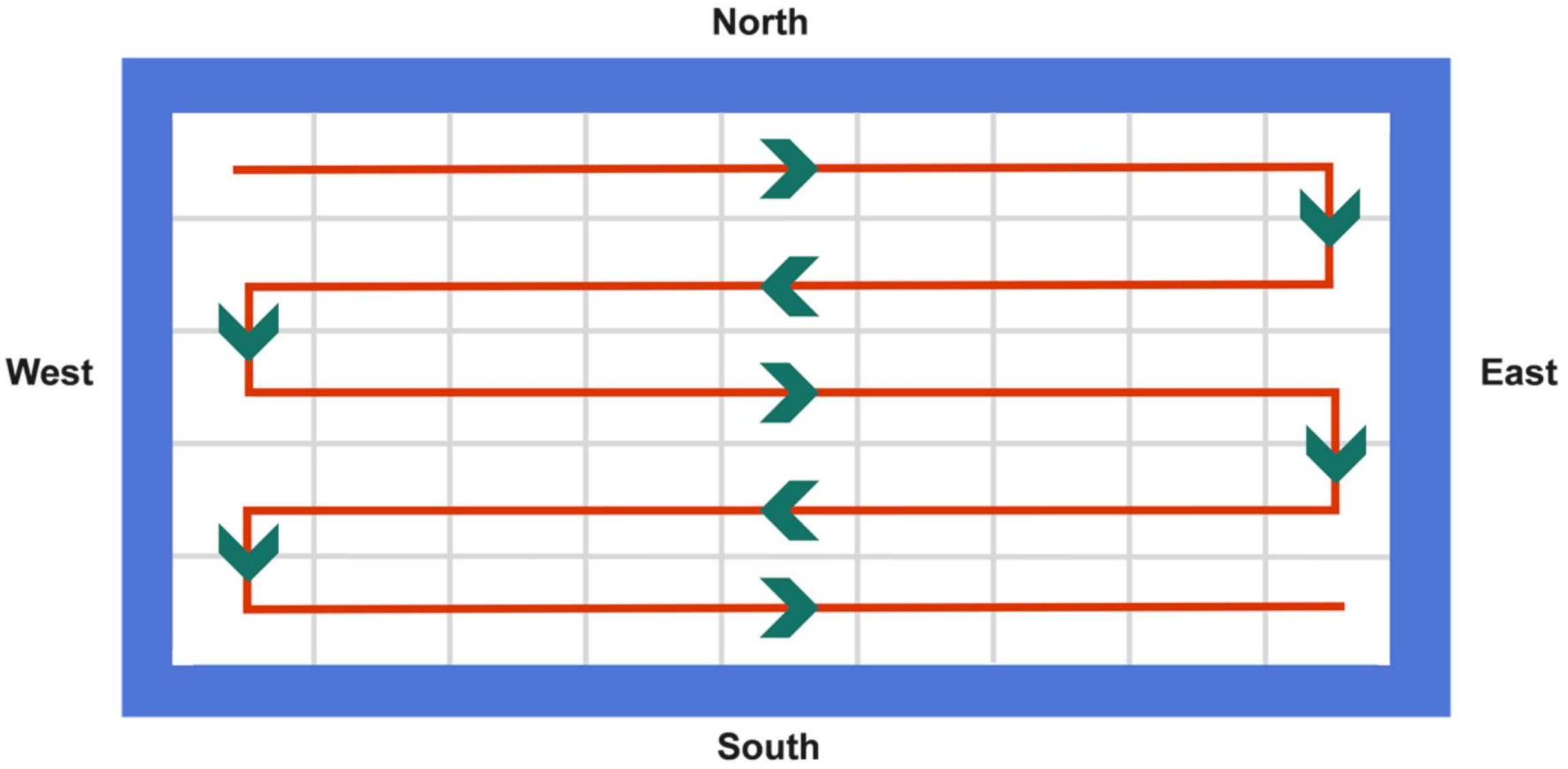
Algorithmic Steps to Solve the Problem

Following are the steps required to solve the problem:

1. Start cleaning from the south-west corner of the pool.
2. Set the direction to north.
3. Set the current direction to north.
4. Move one step ahead.
5. Clean that tile.
6. Repeat steps 4 and 5 until you hit the wall, that is there is a new tile available to clean.
7. Set the direction to east.
8. Move one step ahead.
9. Clean the tile.
10. Set current direction to south if its value is north, otherwise set the current direction to north.
11. Repeat steps 4 to step 10 until you reach the south-east or north-east corner of the pool.

Is there any
alternative solution?

Clean the Pool Starting From the North-West Corner



Algorithmic Steps to Solve the Problem

Following are the steps required to solve the problem:

1. Start cleaning from north-west corner of the pool.
2. Set the direction to east.
3. Set the current direction to east.
4. Move one step ahead.
5. Clean that tile.
6. Repeat steps 4 and 5 until you hit the wall, that is there is a new tile available to clean.
7. Set the direction to south.
8. Move one step ahead.
9. Clean the tile.
10. Set current direction to west if its value is east, otherwise set the current direction to east.
11. Repeat steps 4 to step 10 until you reach the south-east or south-west corner of the pool.

Characteristics of a Good Algorithm

- **Precision:** The steps are precisely stated or defined.
- **Uniqueness:** The results of each step are uniquely defined and only depend on the input and the result of the preceding steps.
- **Finiteness:** The algorithm always stops after a finite number of steps.
- **Input:** The algorithm receives some input.
- **Output:** The algorithm produces some output.

Pseudocode

What Is a Pseudocode?

- A Pseudocode is one of the ways to represent an algorithm. It is a plain language description of the steps in an algorithm.
- It often uses structural conventions of a normal programming language.
- It is intended for human reading rather than machine reading.
- It helps in transitioning to writing programs easily.

Can we now create pseudocode based on the algorithm steps written to clean the pool?

Pseudocode to Clean the Pool

Write a pseudocode based on the algorithm steps written to clean the pool.

DEMO

Pseudocode to Clean the Pool

```
Begin
    Move to the south-west corner
    Clean
    Set direction = north
    Set currentDirection = north
    Set isClear = true
    Repeat
        Repeat
            If isClear
                Move one step ahead
                Clean
            Endif
        Until not isClear
// Hit the wall, move east by 1 tile
        Set direction = east
```

```
//Move if there is space
         If isClear
            Move one step ahead
            Clean
        Else
            Go to End
        Endif
// Now change the traversal direction
        If currentDirection is north
            set direction = south
        Else
            set direction = north
        Endif
    Until not isClear
End
```

Think and Tell

- You saw that a few keywords like move, clean, and set are sufficient to give instructions to program the pool cleaner.
- But to solve a more generic problem by a general-purpose computer, more **keywords** to give instructions may be needed.
- Pseudocode uses some reserved words to write algorithms. These include:
  - Input/Get
  - Compute/Calculate
  - Print
  - If-Else
  - While
  - True-False
  - Repeat-Until

Benefits of a Pseudocode

- Pseudocode can be quickly and easily translated into an actual programming language as it is similar to a programming language.
- Even non-programmers can understand it easily.
- It clearly conveys the meaning that it is intended for, even if it has some syntax errors.
- Pseudocode makes it quite easy to incorporate any changes to the design.

Can we now try some examples to solve a trivial problem that gives output on providing inputs?

Area of a Rectangle

- **Area of a Rectangle:** Use the formula length*width to calculate the area.

Step 1: Start
Step 2: Accept the length and width from the user
Step 3: Calculate the area using the formula length*width
Step 4: Display area
Step 5: Stop

Algorithm: Simple English Instructions for Humans

```
BEGIN
    GET length, width
    CALCULATE length*width and store in area
    PRINT area
END
```

Pseudocode: Precise Instructions for Computers

Check for Odd or Even Number

- **Even Number Definition:** A number is even if it is fully divisible by 2.

```
Step1: Start
Step2: Accept a number
Step3: Divide that number by 2
Step3: If the remainder is zero, then it
is evenly divisible and hence displays
as an "even number"
Step4: If the remainder is nonzero, then
is it is not divisible and displays as
"Odd number"
Step6: Stop
```

Algorithm: Simple English Instructions for Humans

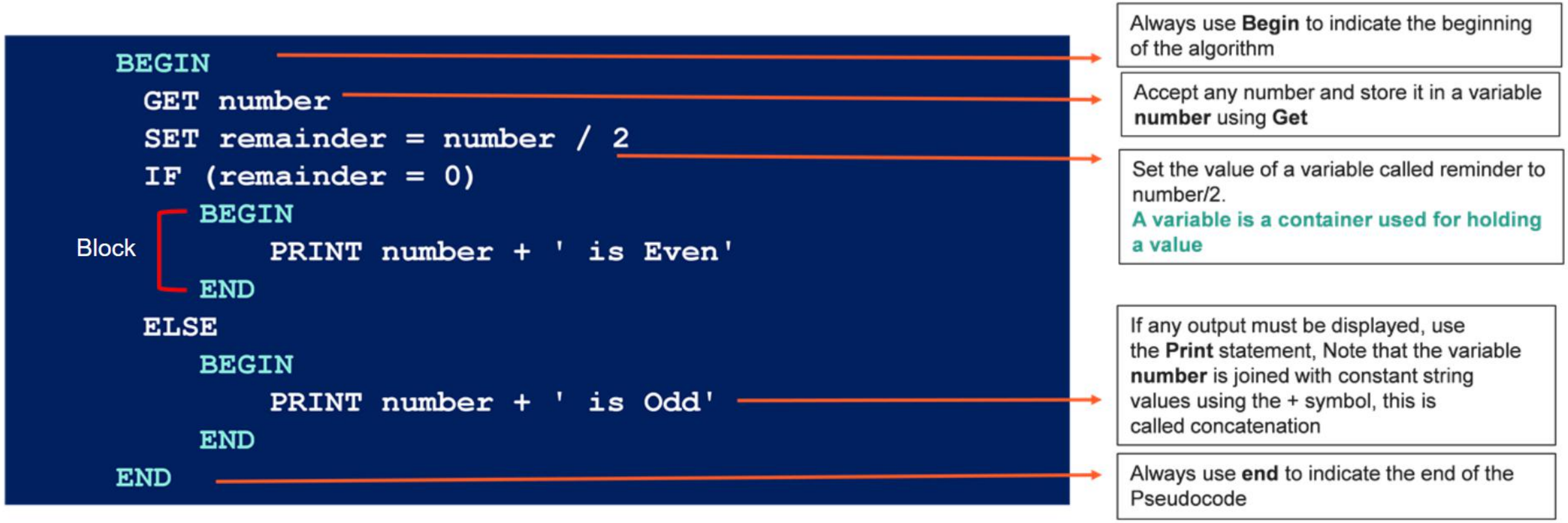
```
BEGIN
  GET number
  SET remainder = number / 2
  IF (remainder = 0)
     BEGIN
          PRINT number + ' is Even'
     END
  ELSE
     BEGIN
          PRINT number + ' is Odd'
     END
END
```

Pseudocode: Precise Instructions for Computers

How Are These Instructions Processed by Computer Systems?

- All modern computers function on the general model of input, process, and output.
- A computer system:
  - Accepts inputs from the user
  - Processes the input
  - Generates the output

Keywords Used in Pseudocode



- In the selection structure, to check for a condition, IF is used,
  - If the condition is true, the block of IF is executed.
  - If the condition is false, the block of ELSE is executed.

Think and Compare

The **swimming pool cleaner** problem is a non-trivial problem that involves logical thinking and then instructions are provided in a language that can be understood.

Checking whether a number is even or odd is a trivial problem that is quite straightforward and where the solution is known from its definition. To solve, the instructions should be provided in a language that the computer can understand.

Calculate Sum of Natural Numbers

- Given a maximum limit, find the sum of all the natural numbers till the maximum limit.

```
BEGIN
  GET max_limit
  SET sum = 0
  SET counter = 0
  FOR counter = 1 TO max_limit STEP 1
    DO
          sum = sum + counter
    ENDFOR
  PRINT sum
END
```

Calculate the Time Required to Complete this Journey

Calculate the time taken to reach the journey when the input values for the speed and distance to be covered are provided by the user.

Write an algorithm and pseudocode for the above problem.

Formula: Speed = Distance/Time
Time = Distance/Speed

Example: Speed = 70 miles/hour
Distance = 280 miles
Time Taken = 280/70 = 4 Hours

DEMO

Algorithm vs. Pseudocode

Algorithm	Pseudocode
An algorithm design technique is a general approach used for solving problems algorithmically that is applicable to a variety of problems from different areas of computing.	Pseudocode is written in a format that is similar to the structure of a high-level programming language which can later be converted to a program code easily.
Algorithm uses English statements.	Pseudocode also uses English statements, but they are more relevant to a programming language.
The words used in an algorithm need not be very specific. For example: Accept a number can also be written as Get a number.	The words used are more specific. For example, GET number cannot be modified. Here GET is a keyword which must be used as is.
It is not mandatory to write start and end for an algorithm.	It is mandatory to write blocks of pseudocode between BEGIN and END.