

# Problem Solving Introduction 🚀

# Introduction

- Basics of Problem Solving
- Flowcharts
- Psuedocode



# About Me

Prateek Narang

I ❤️ teaching code and

...

Co-founded Coding Blocks

Co-founded Coding Minutes

Google Engineer

Scaler Instructor



# Flowcharts

- Diagrammatic representation illustrating a solution to a given problem.
- Allows you to break down any process into smaller steps and display them in a visually appealing way.

# Flowchart Components



Start

Terminator

# Flowchart Components



Start

Read N

Terminator

Input/Output

# Flowchart Components

Start

Terminator

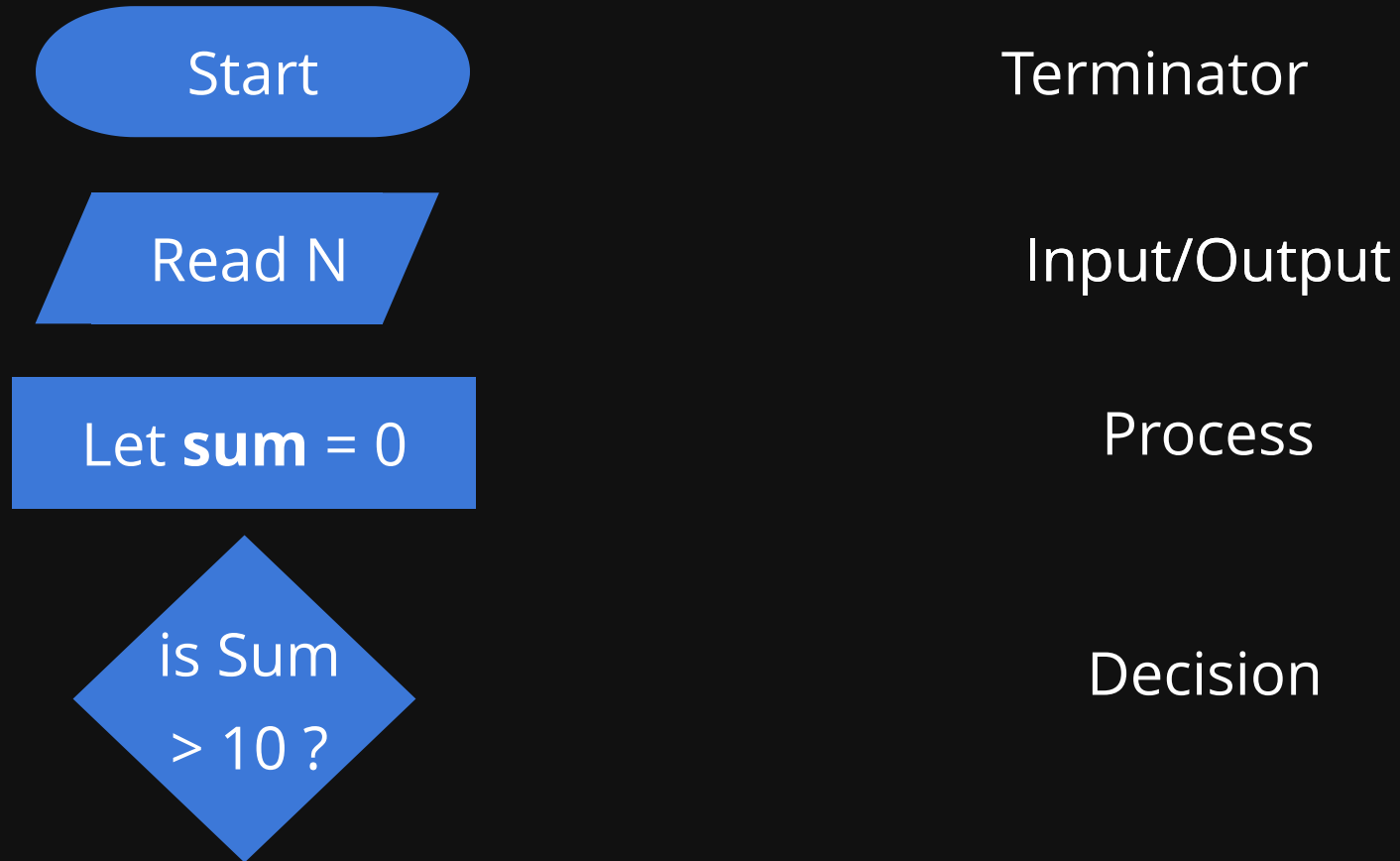
Read N

Input/Output

Let **sum** = 0

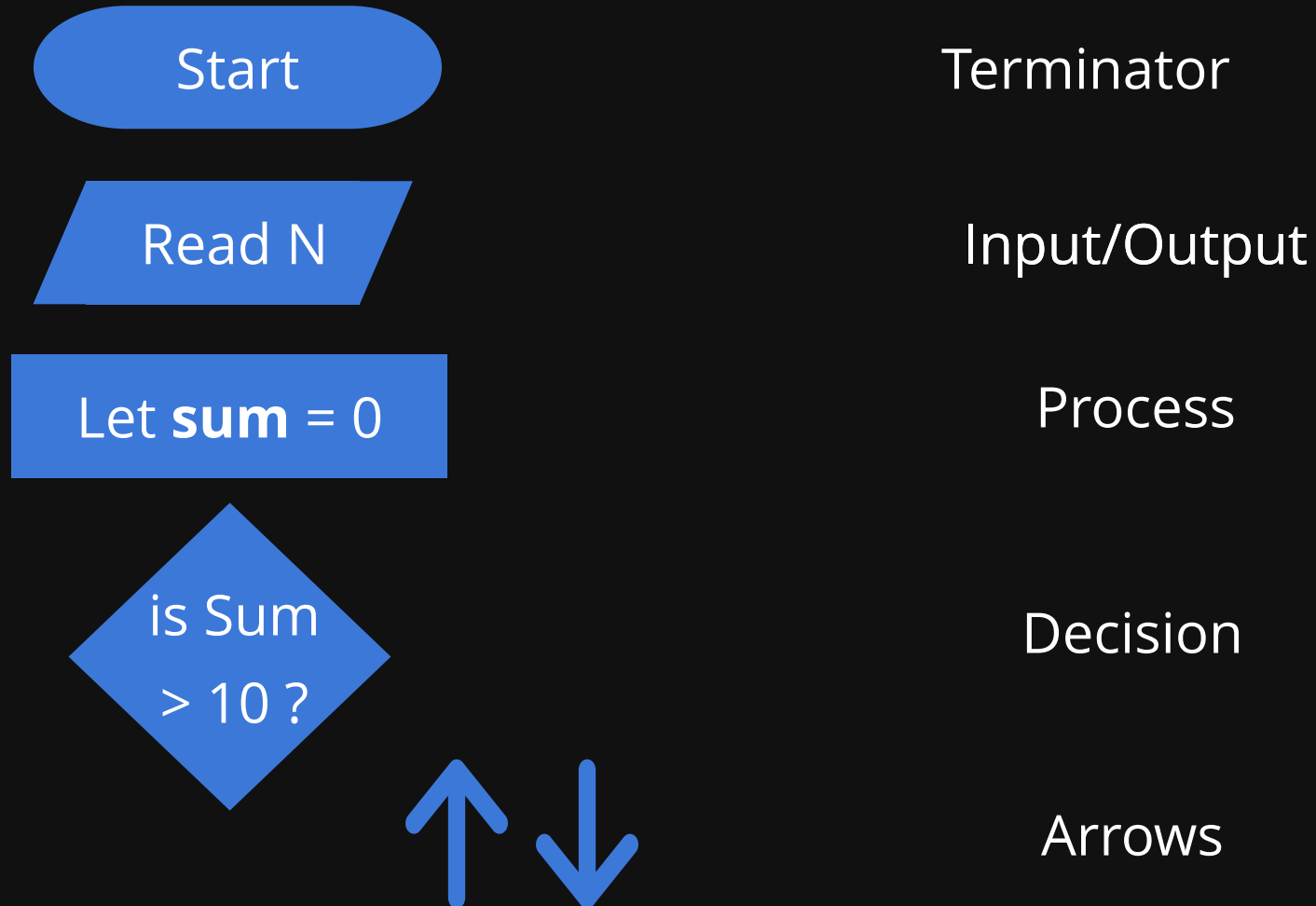
Process

# Flowchart Components





# Flowchart Components



# Examples

Lets look at few problems  
and their flowcharts





# Simple Interest Calculator

Read Principal, Rate & Time and print Simple Interest.

## Sample Input

$P = 100$

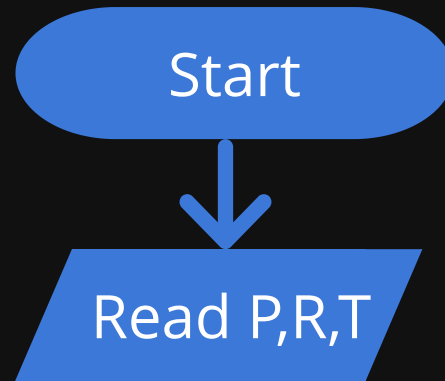
$R = 5$

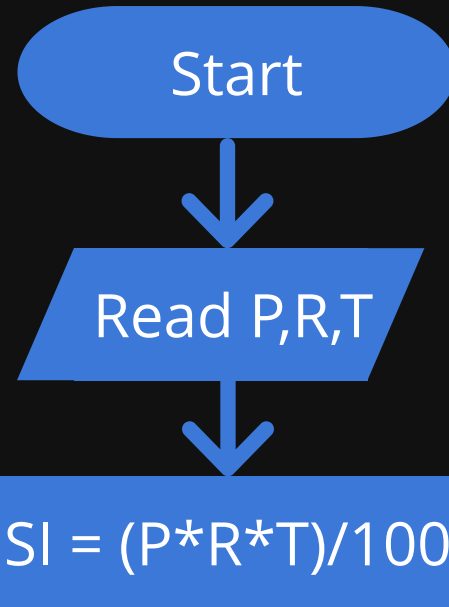
$T = 2$

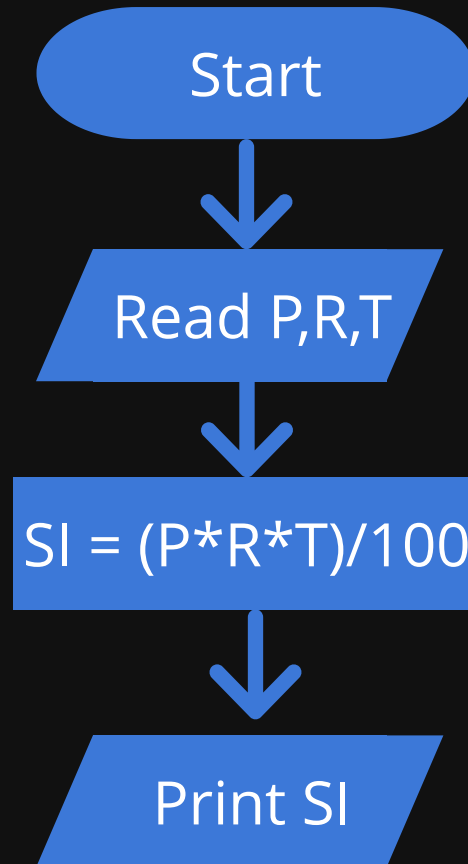
## Sample Output

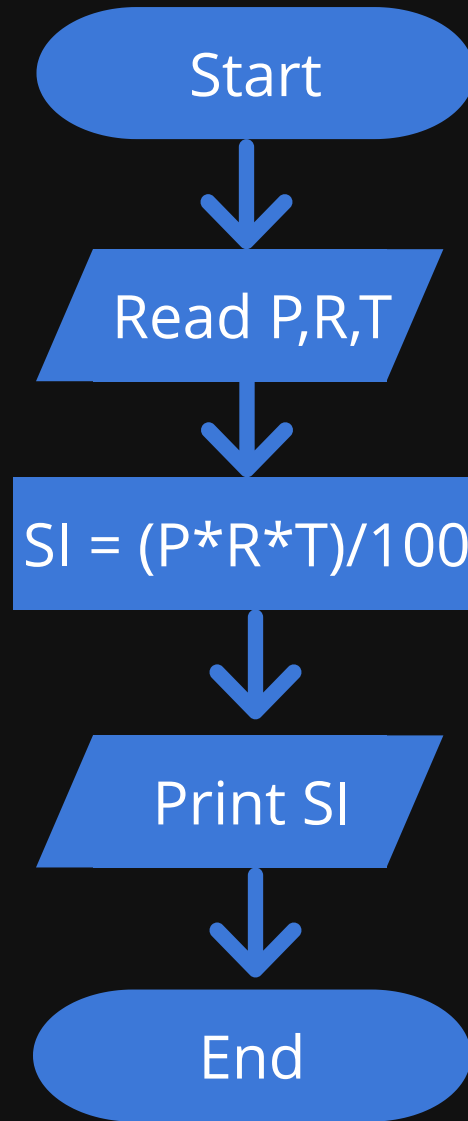
$SI = 10$

Start













# Largest Number

Given 3 numbers, find the largest number.

## Sample Input

A = 10

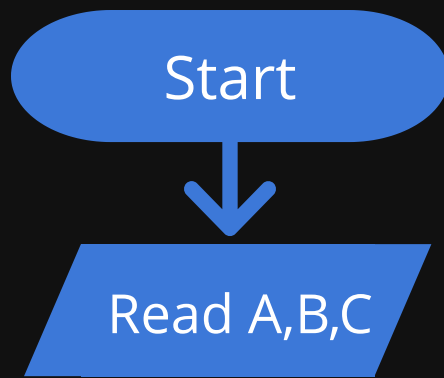
B = 30

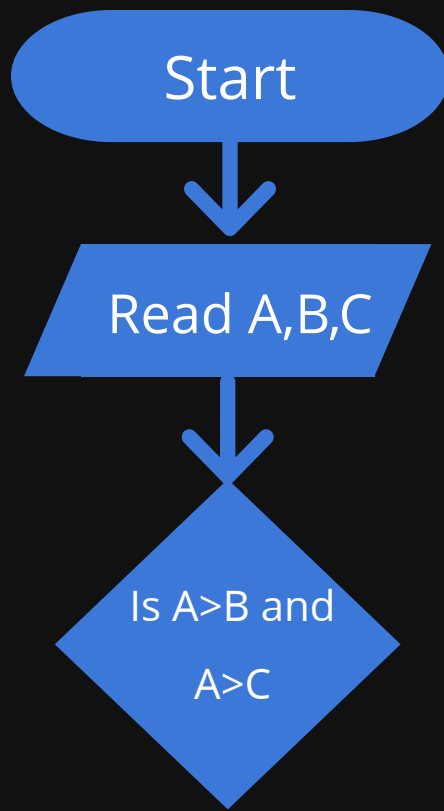
C = 20

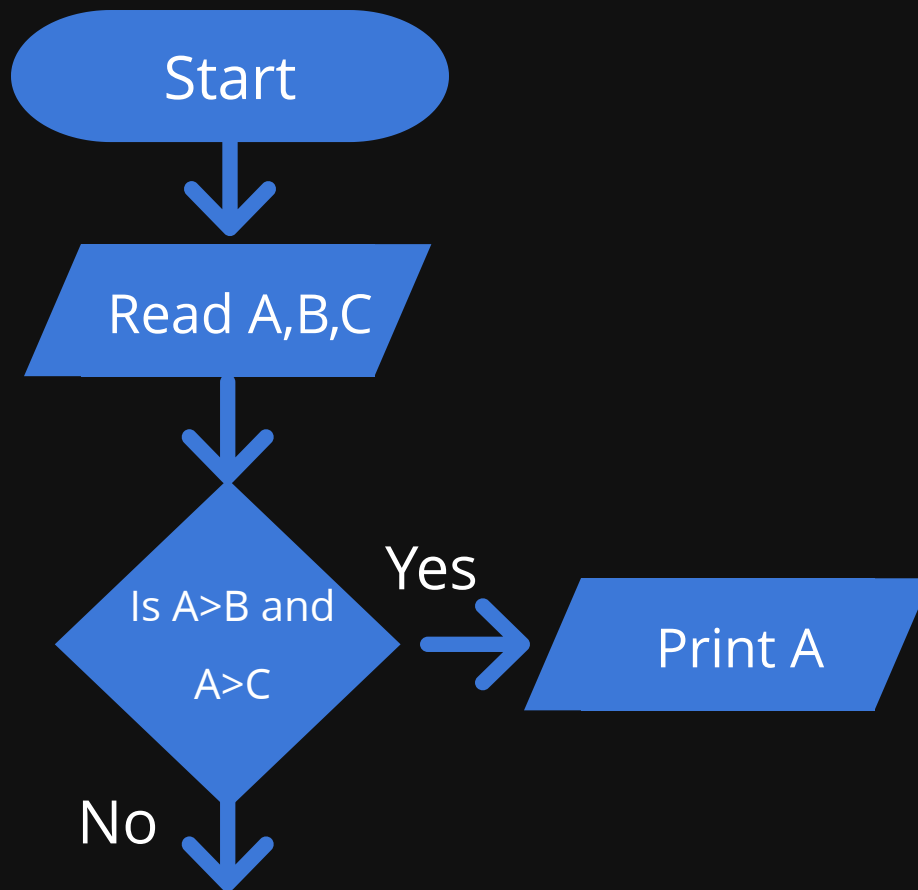
## Sample Output

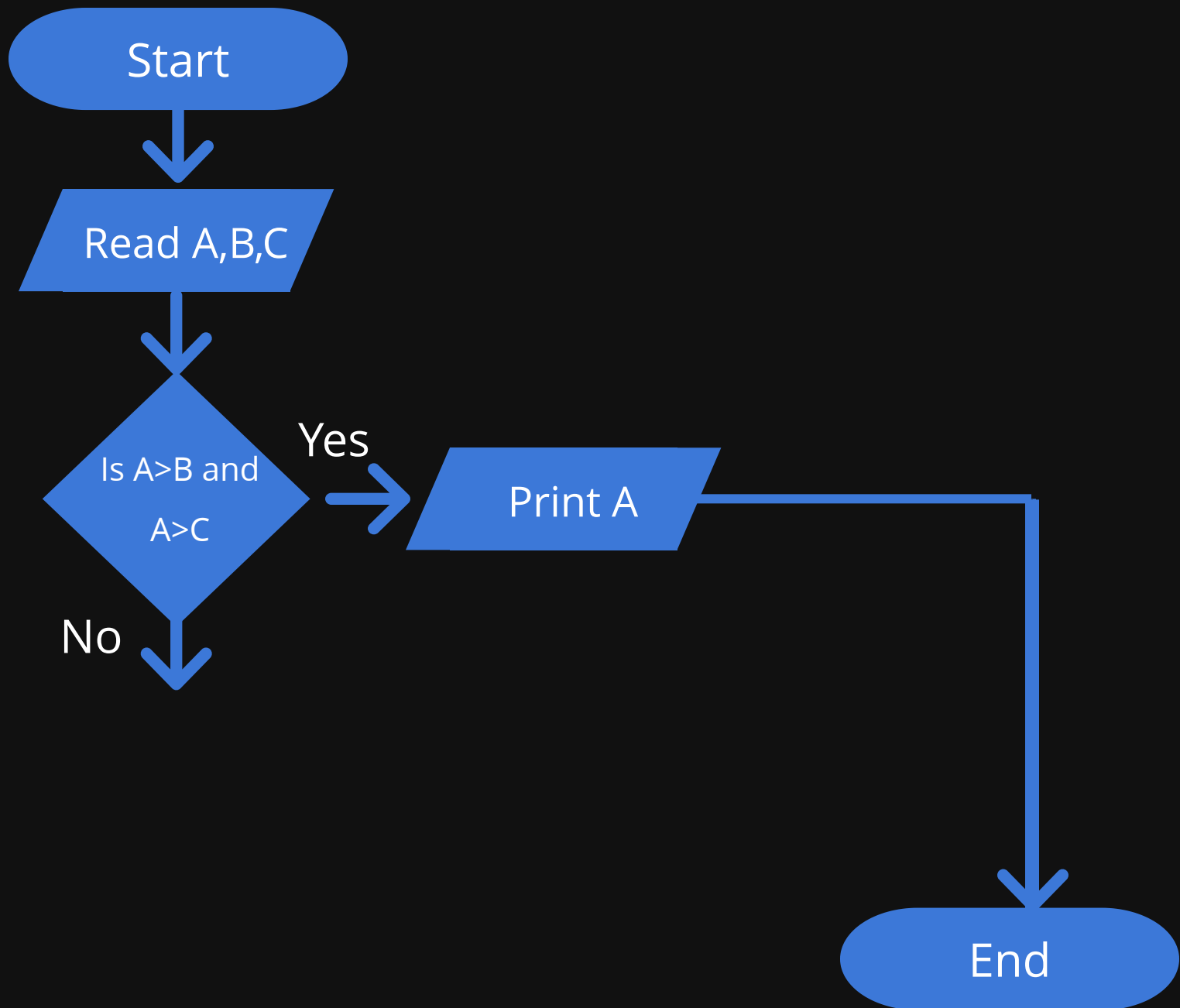
30

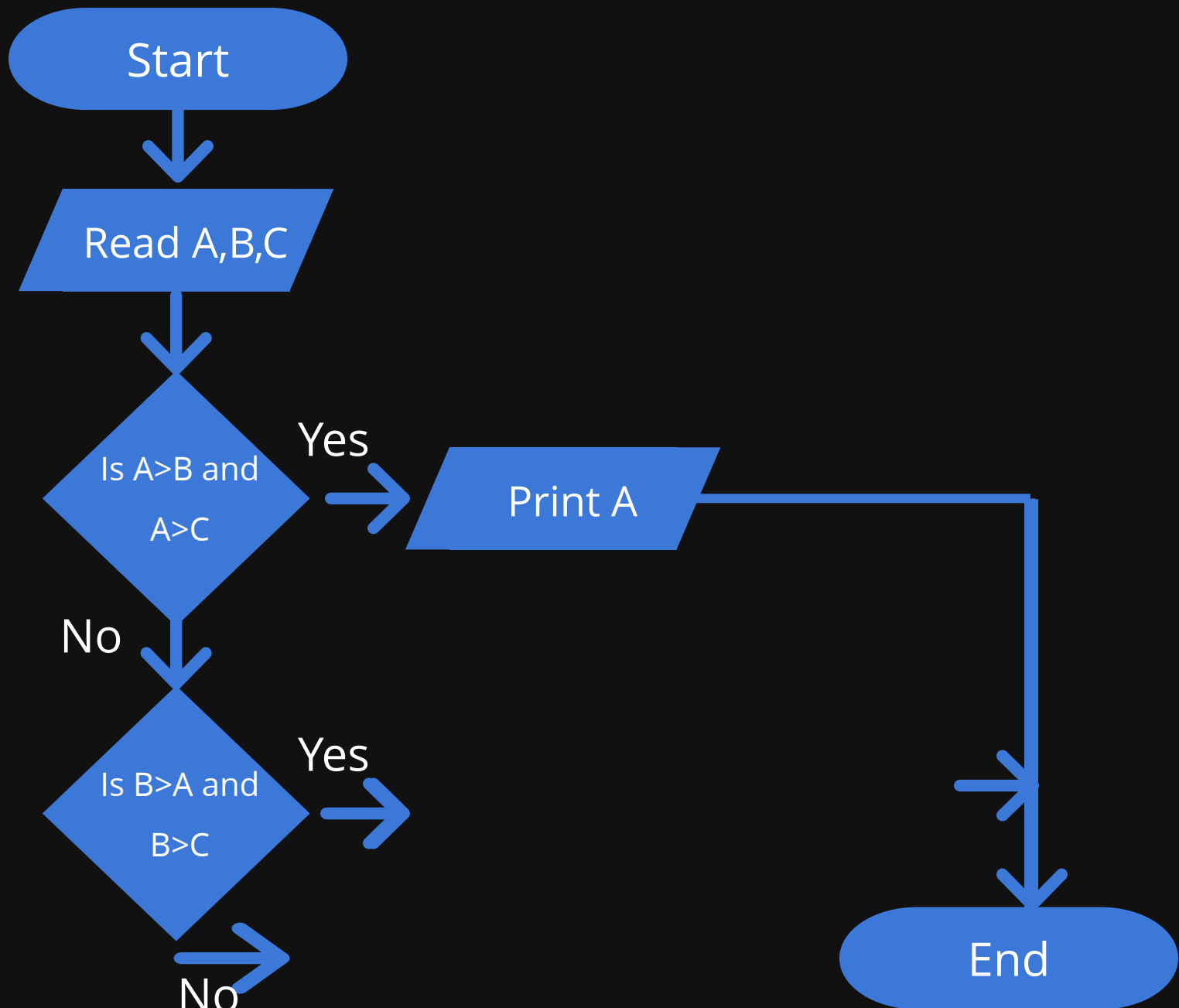
Start

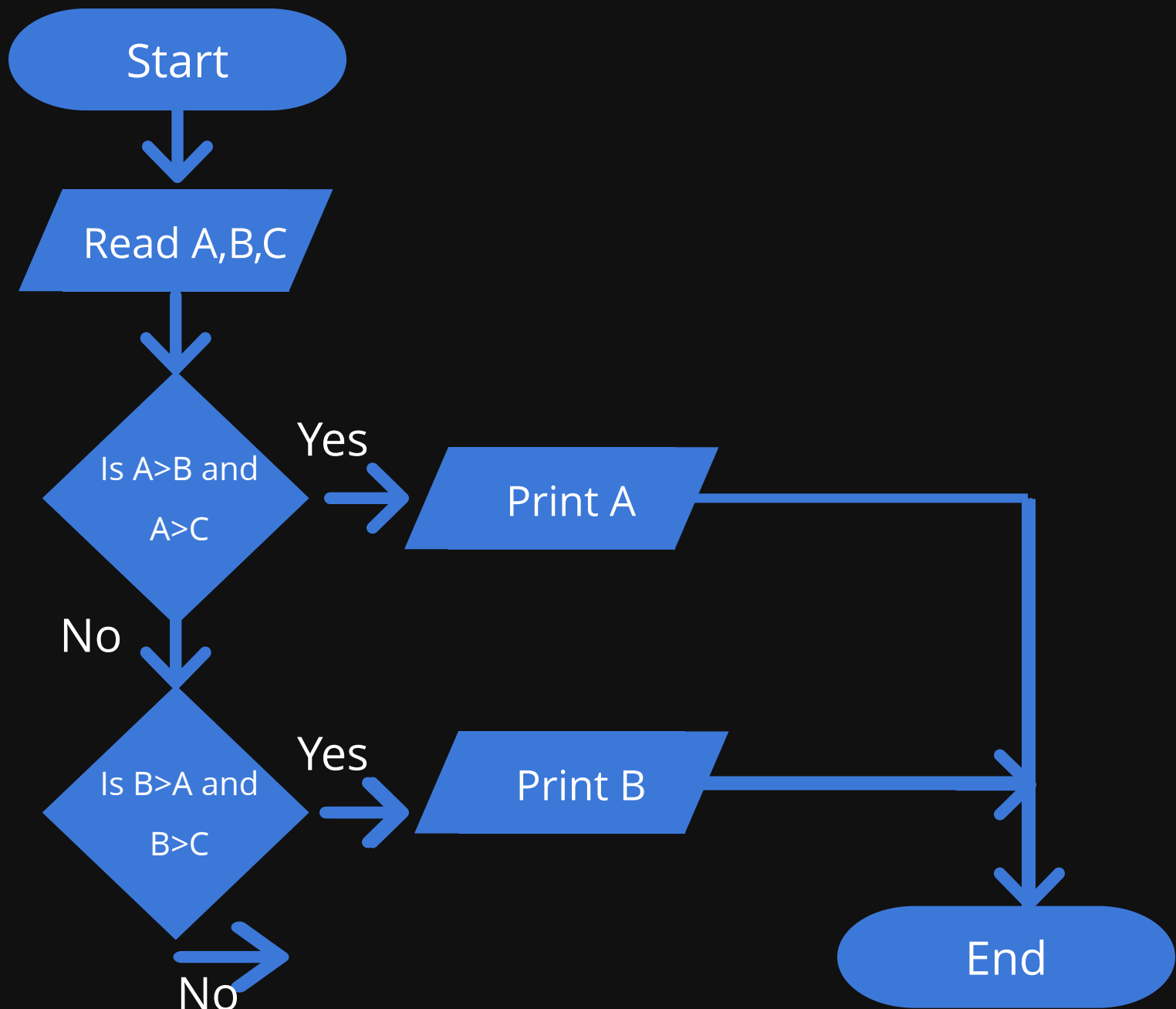




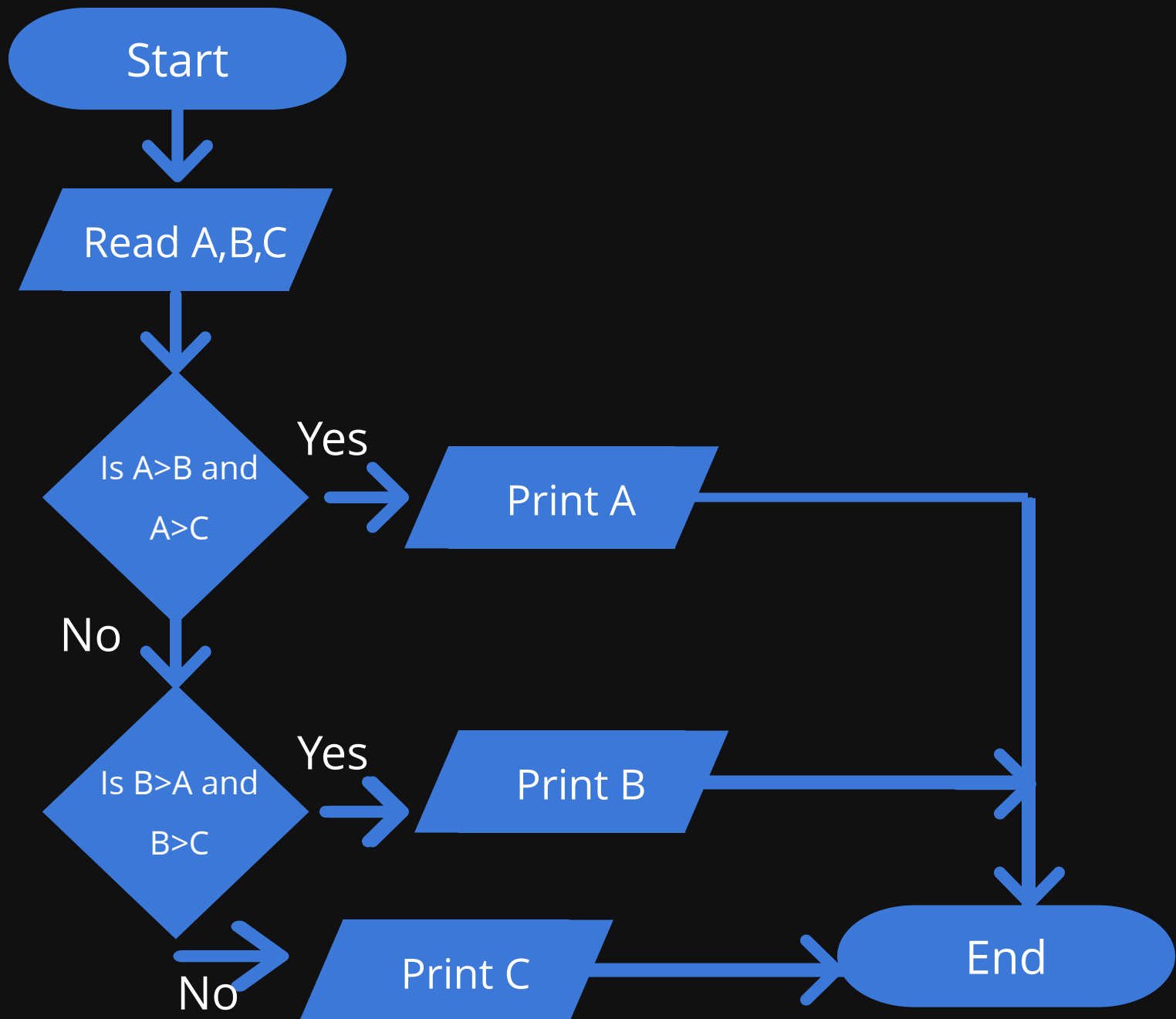














# Number Sum

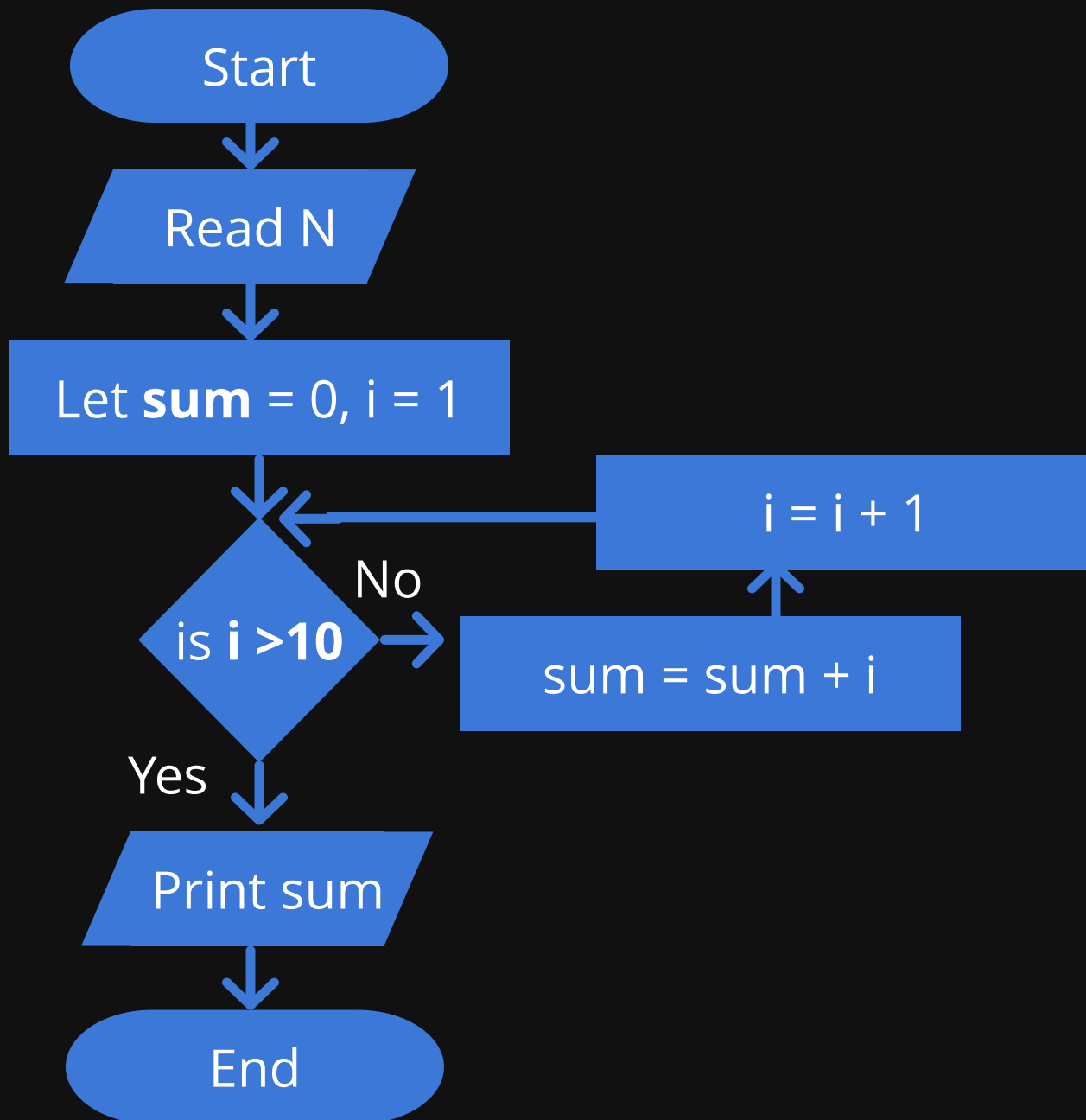
Find the sum of numbers from 1 to N

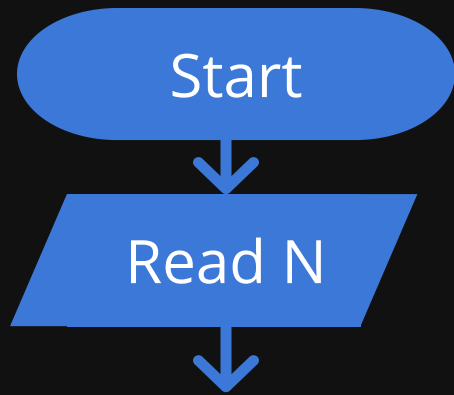
**Example**

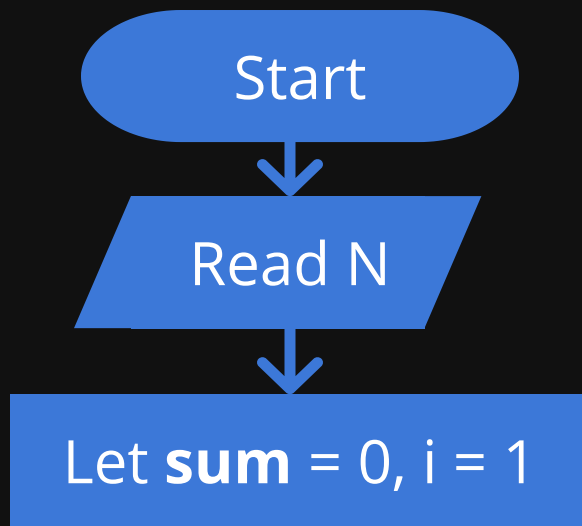
$N = 4$

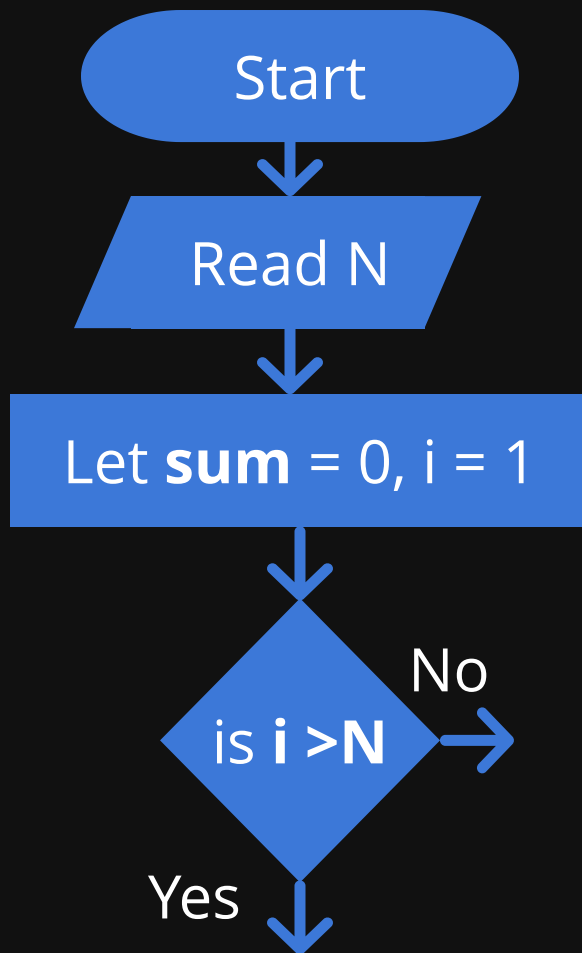
**Output**

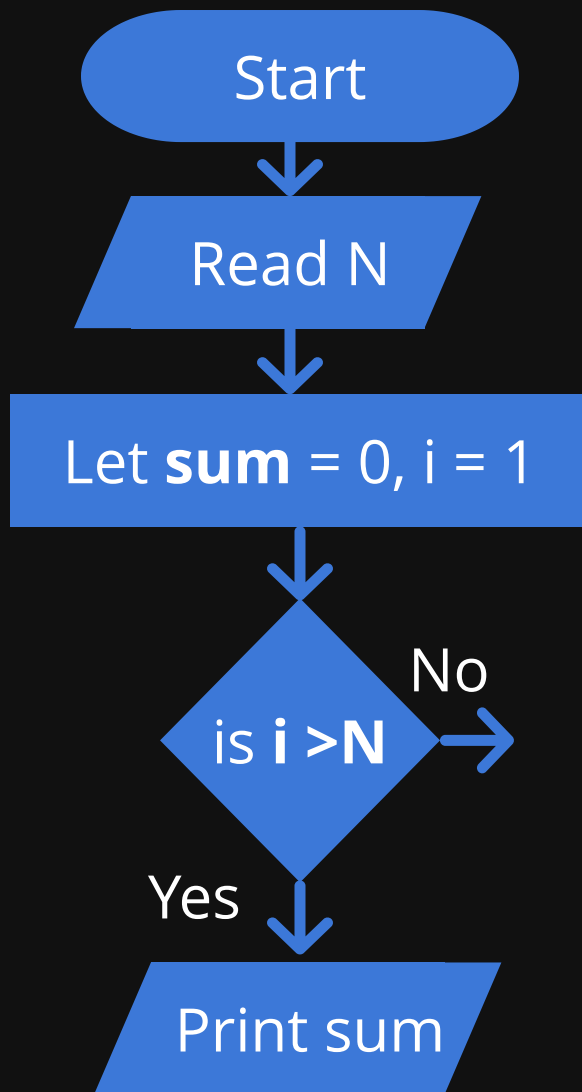
10

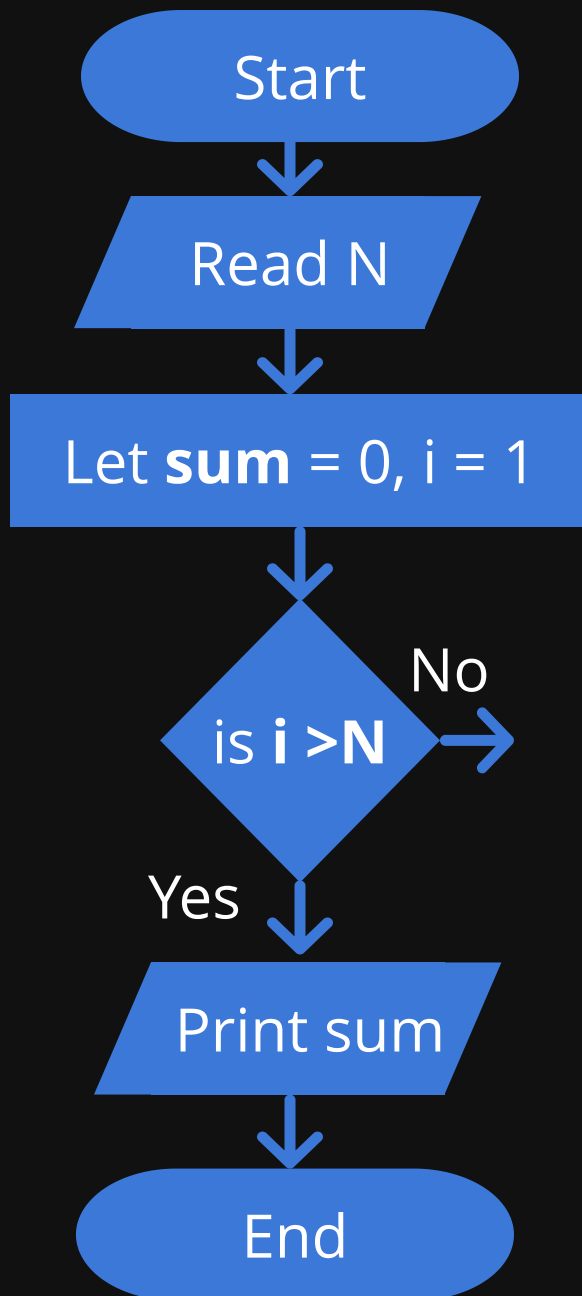




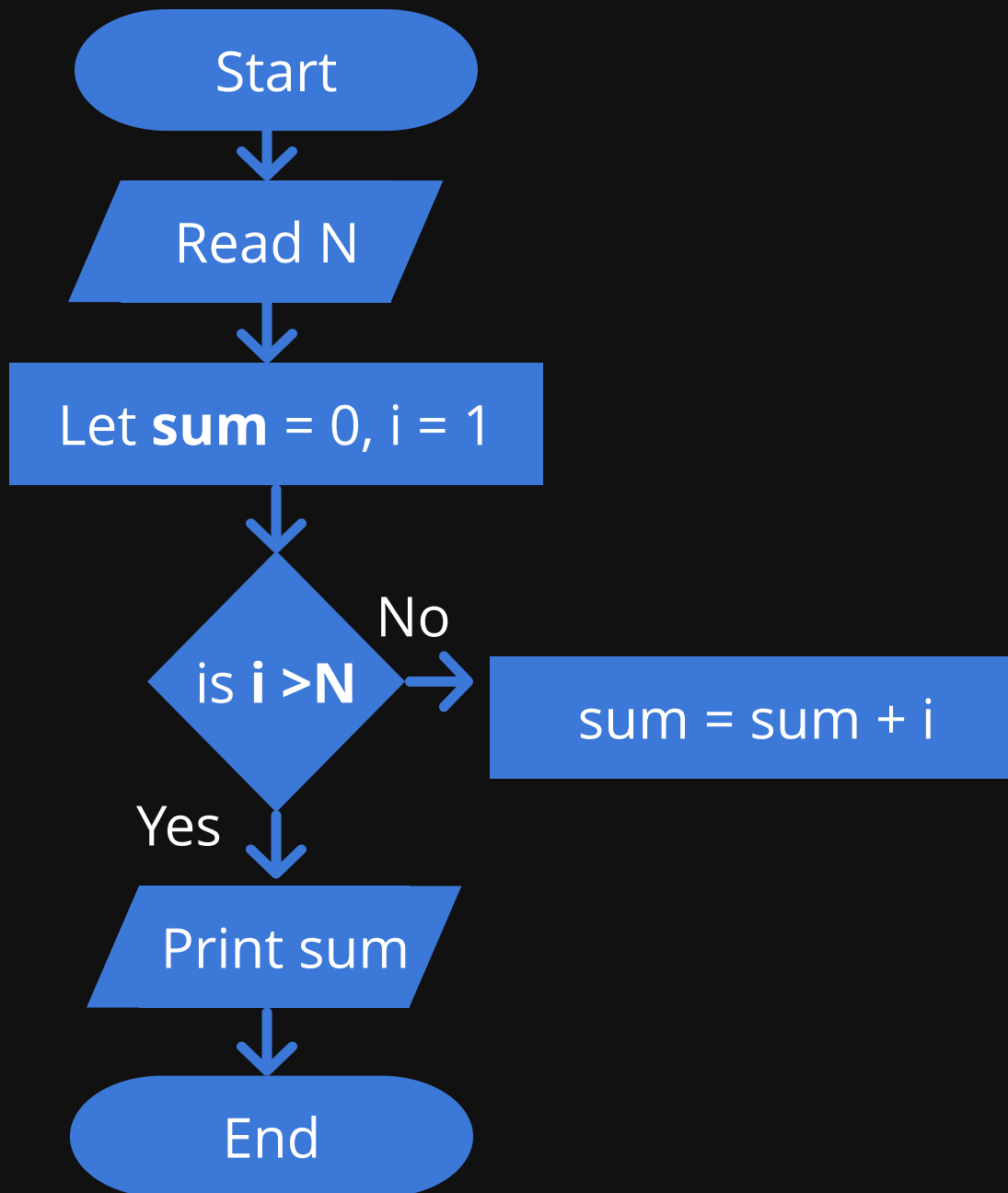


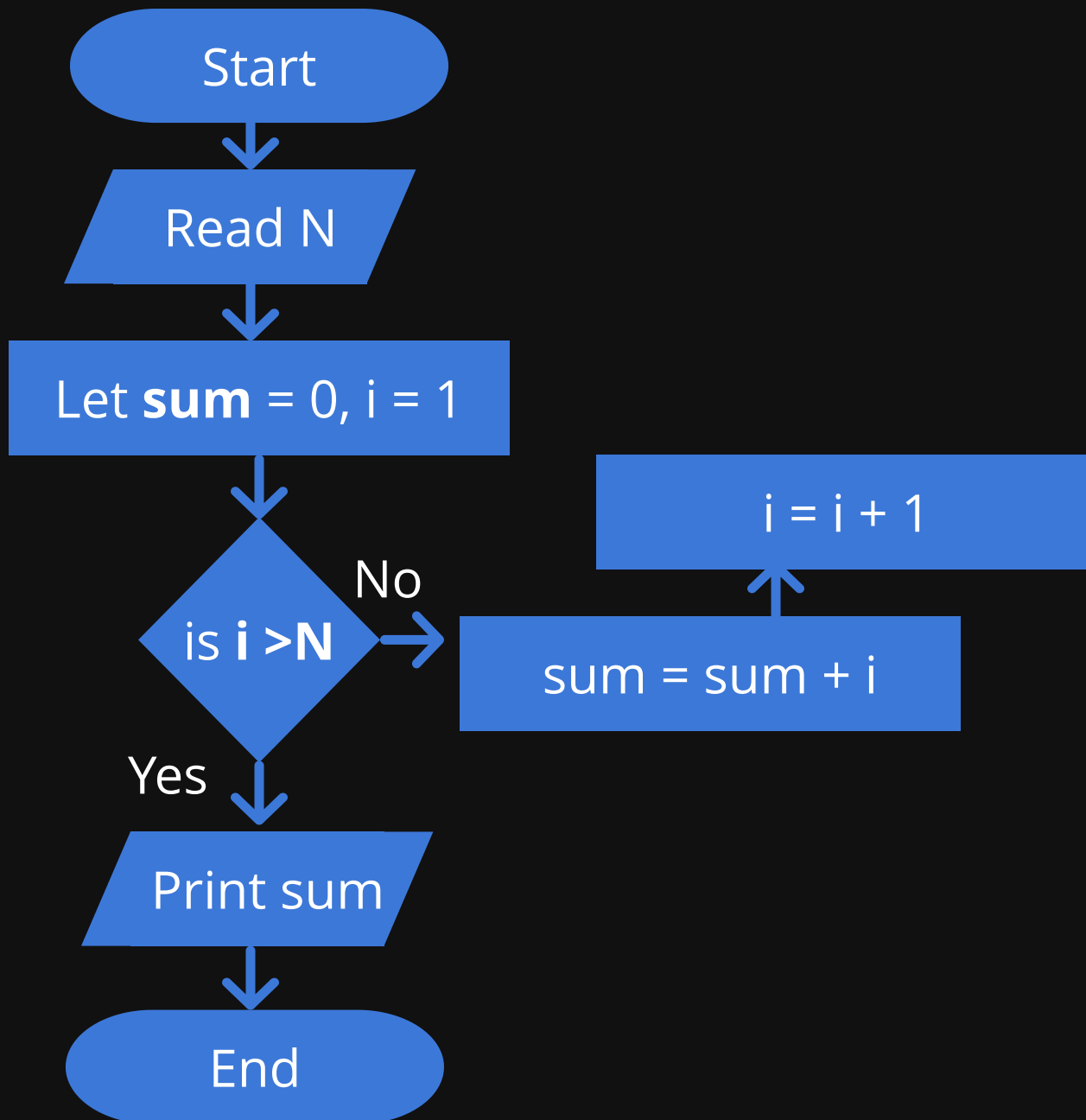


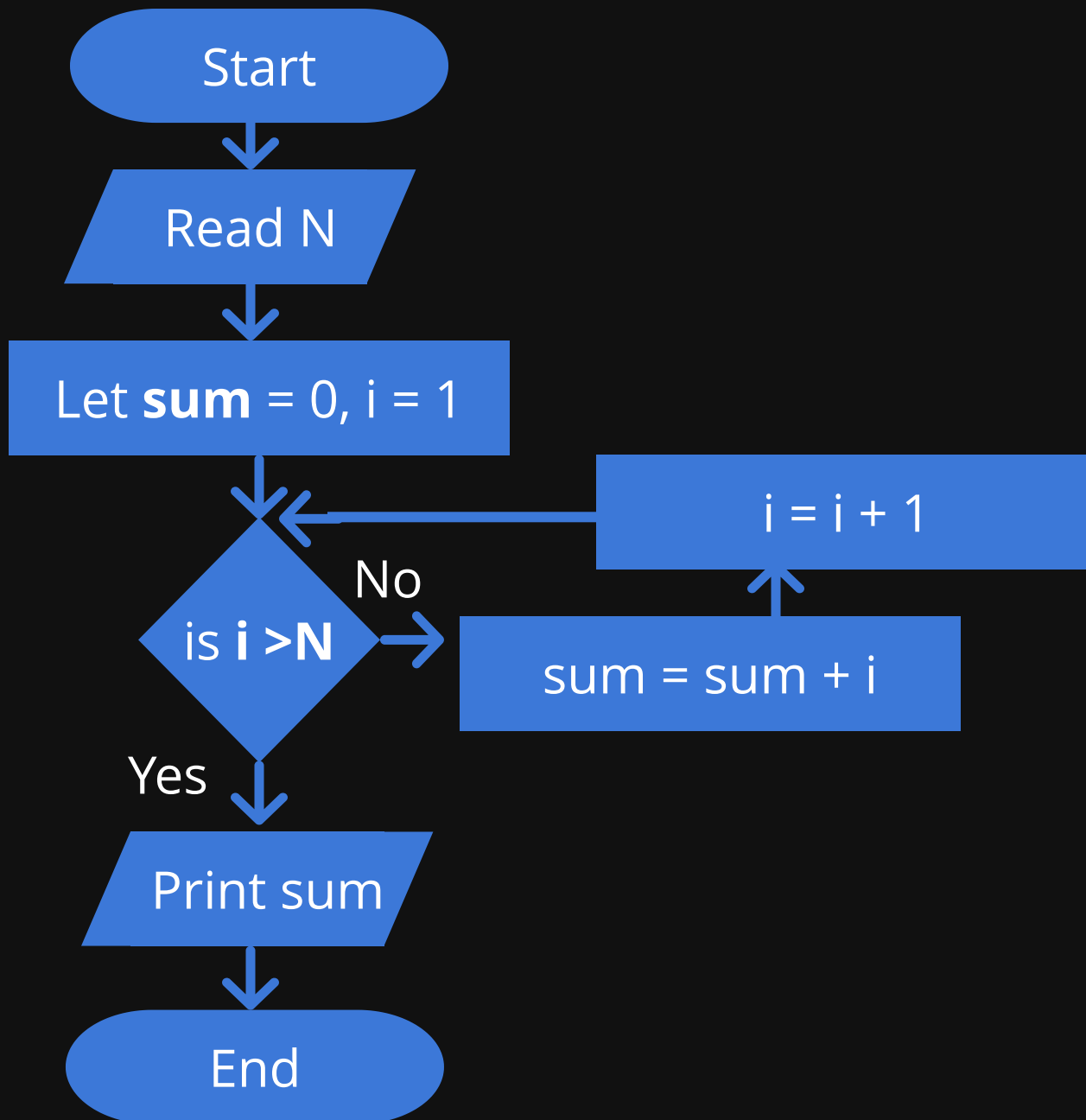














# Time To Try!



**8 Mins**



# Prime Number

Given a Number, check if it is Prime or Not!

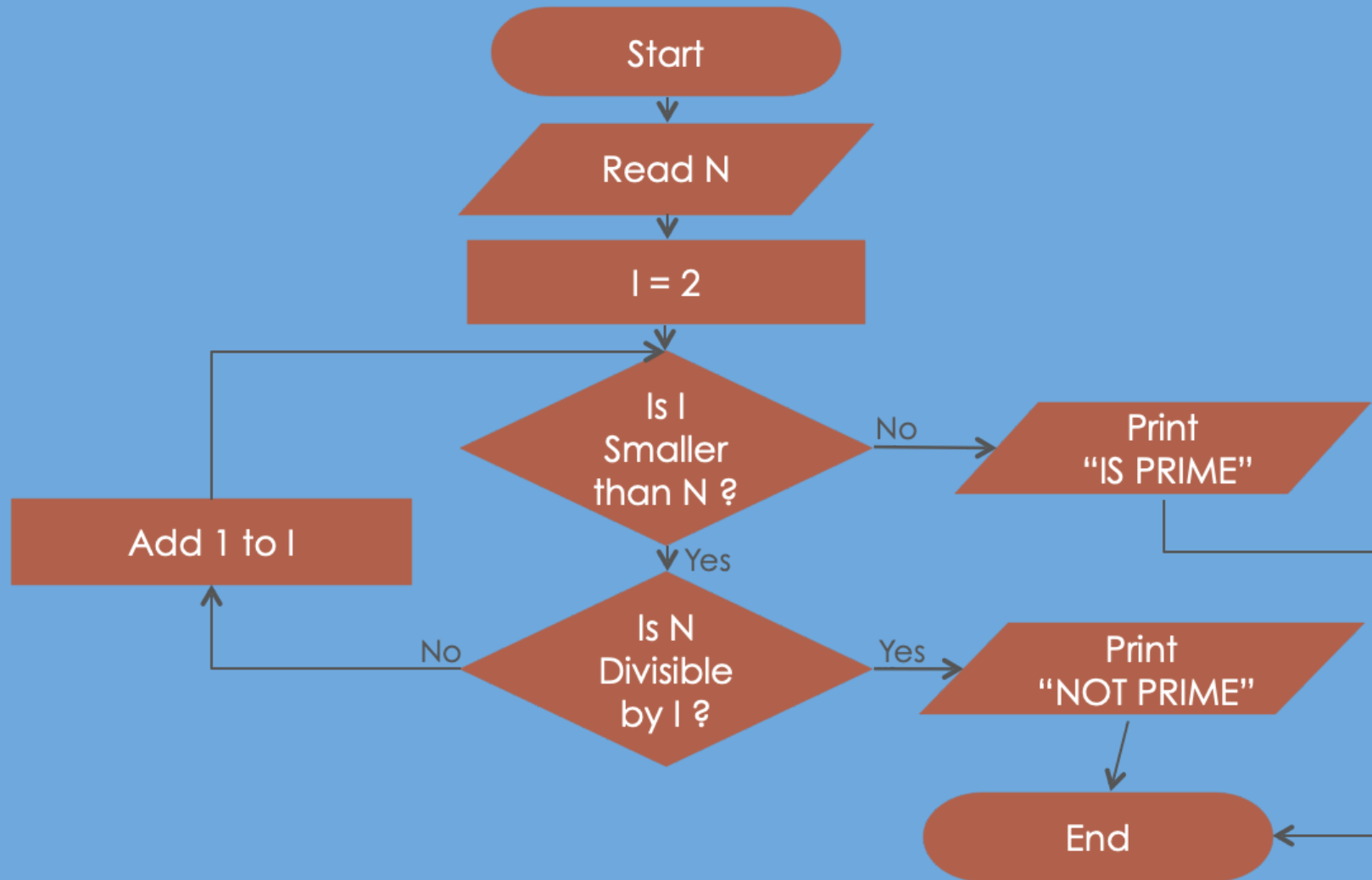
**Input**

11

**Output**

Yes

Let's Try



# A Real World Example!







# Bank

A Bank is open till 6 p.m. and the banker needs to process requests from customers

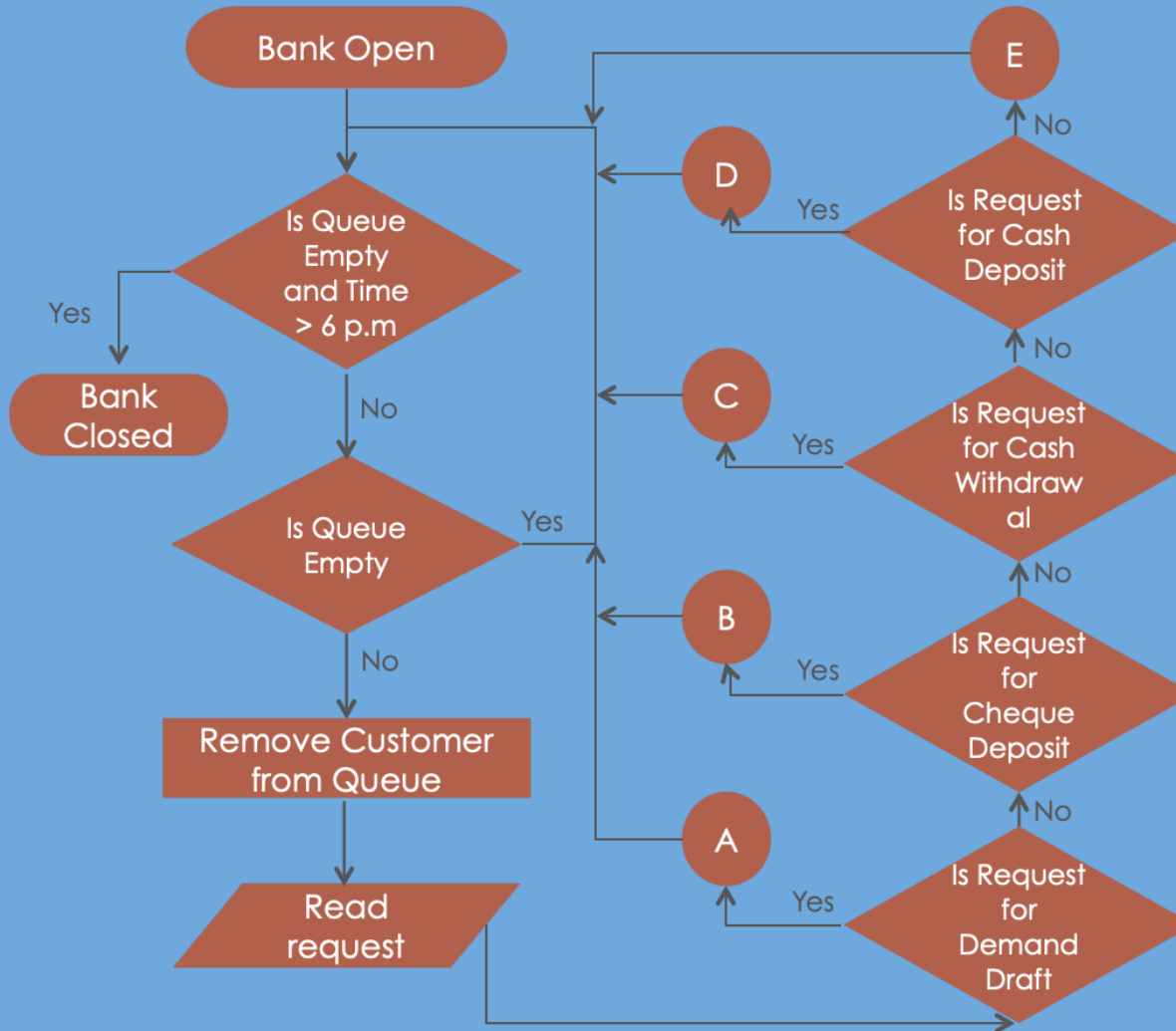
Requests can be one of the four types:

- Cash Deposit
- Cash Withdrawal
- Cheque Deposit
- Making A Demand Draft

Draw the workflow of the bank employee.

Let's Try

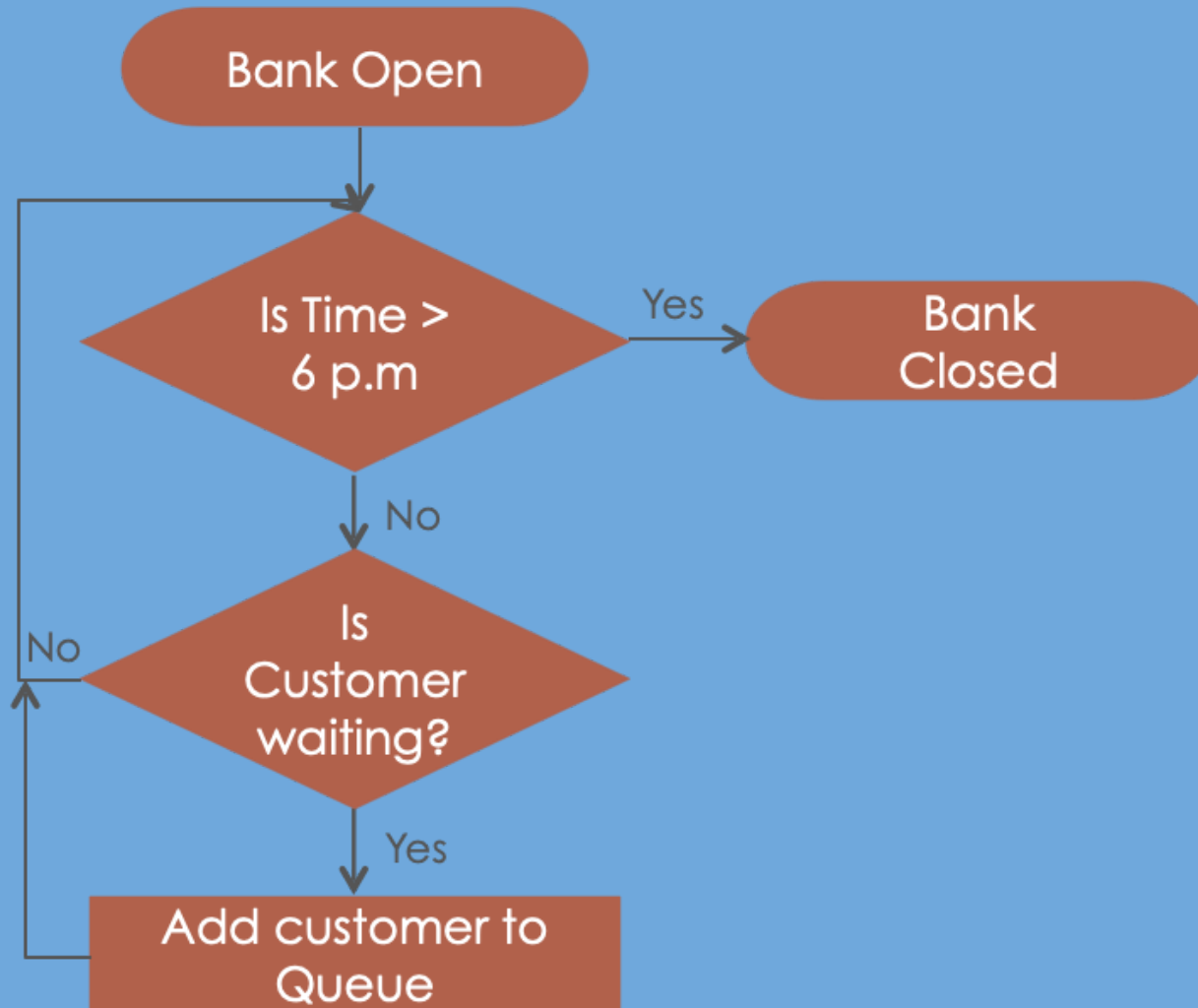
# Solution



## Flowchart for **Bank Security Guard?**

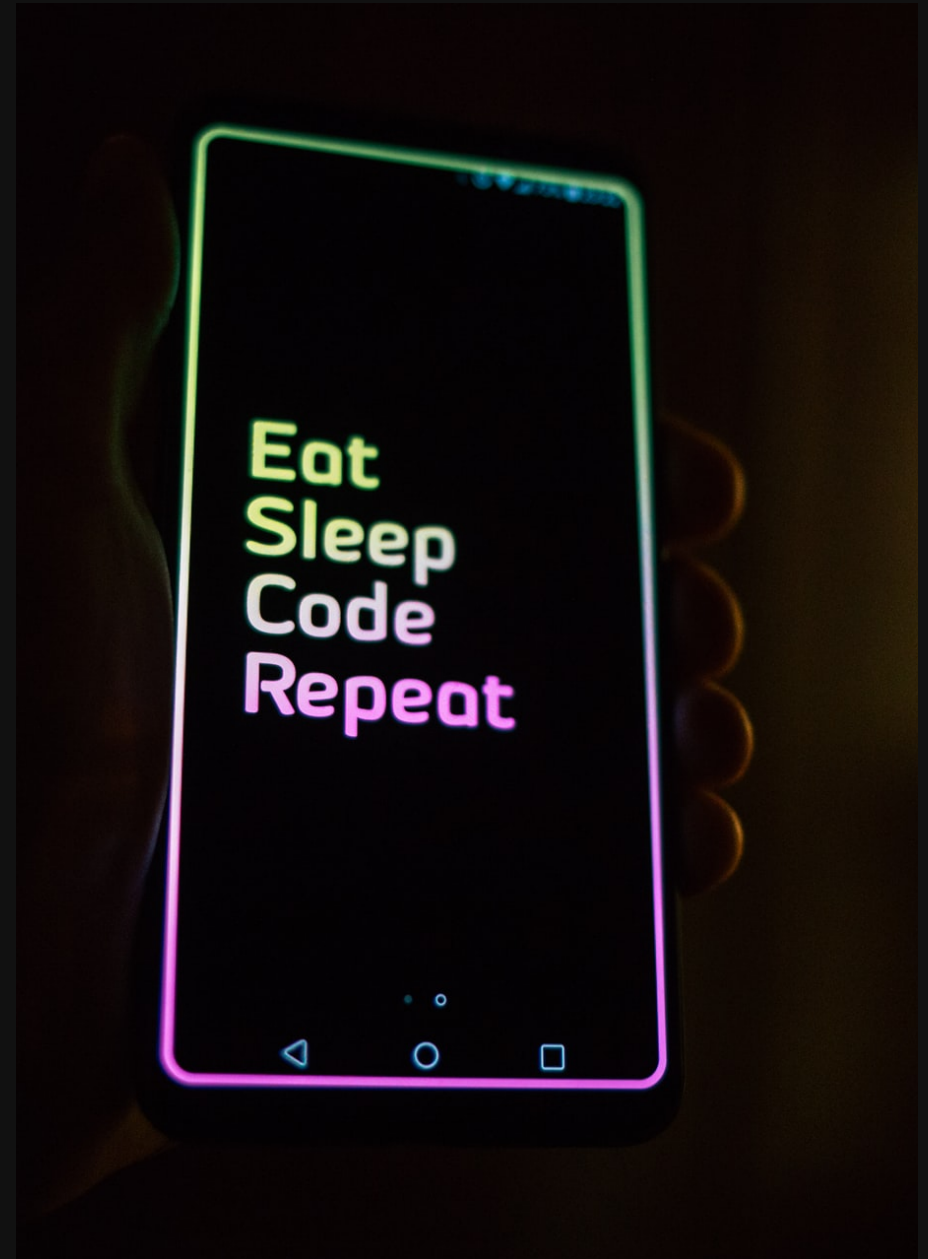


# Solution



# Pseudocode

Human readable description  
of an algorithm



# It's helpful!

- Language Independent.
- Structure your code before writing it.
- Fastest way to verify / get a review.



```
2 const fetch = require('...')
3 const log = require('log')
4 let embed;
5
6 function transformIt, ...transformations
7 // Promise.resolve to promise: ...
8 return transformations.reduce((prev, trans) => {
9   // ...
10 }, Promise.resolve(embed));
11
12 function removeWhiteSpace(prev) {
13   return prev.then(() => {
14     $('header').empty();
15     const children = $('header').children();
16     if (children.length) {
17       $(header).append(children);
18       $(children).remove();
19     }
20     return header;
21   });
22   return Promise.resolve();
23 }
```



# Pseudocode

Let's define our own instruction set

- **Input** [ read N ]
- **Assignment** [ Sum = 0 ]
- **Output** [ print Sum]
- **If Else**
  - [ if  $I < N$  then ... end else then ... end ]
- **While Loop**
  - [ while  $I < N$  do ... end ]
- **Exit** [exit]





# Prime Number

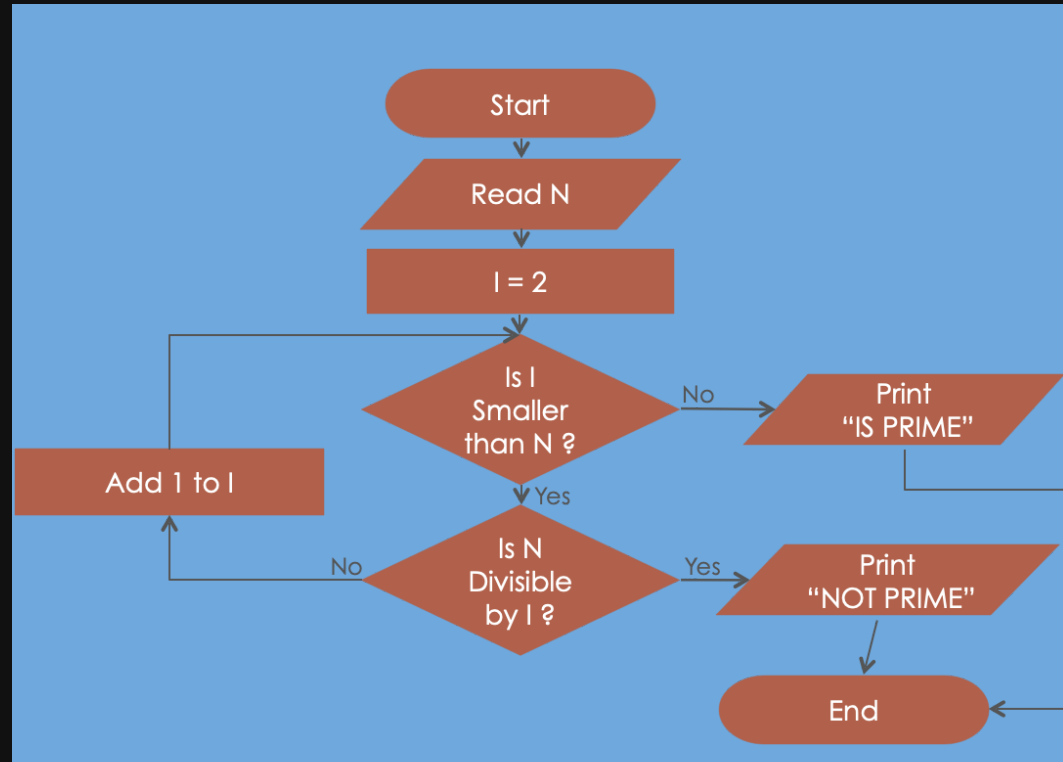
Pseudocode to check if number is prime or not.

## Input

11

## Output

Yes



```
read N
```

```
i = 2
```

```
while i < N do
```

```
    if N is divisible by i then
```

```
        print "NOT PRIME"
```

```
    exit
```

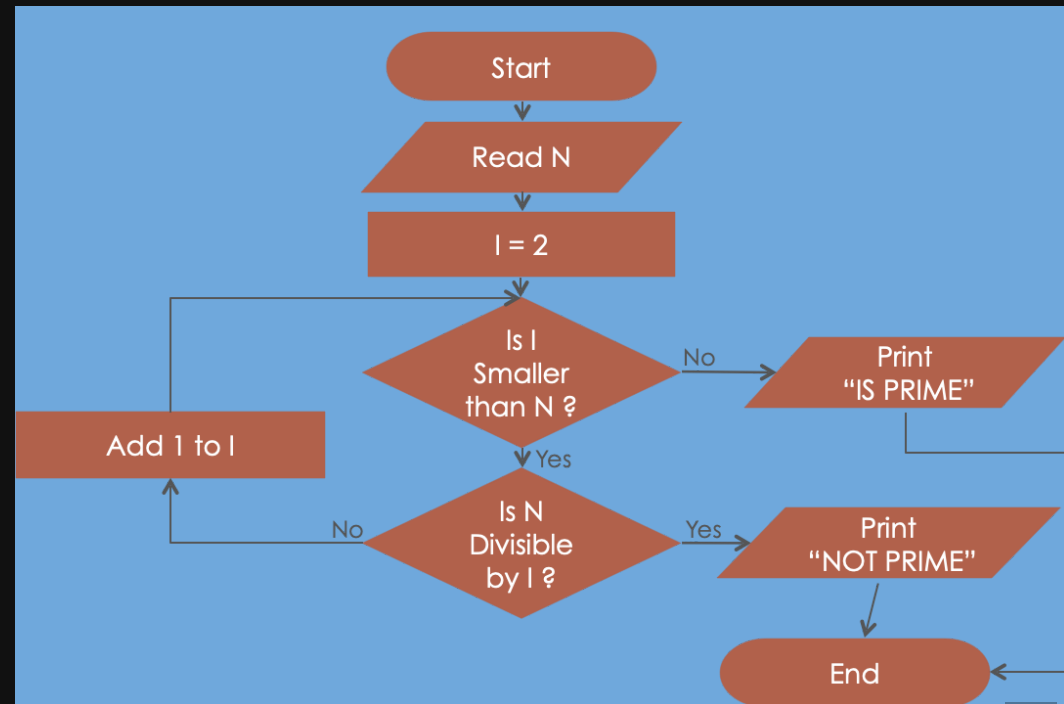
```
end
```

```
    i = i + 1
```

```
end
```

```
print "IS PRIME"
```

```
exit
```





# Number Pattern

Print the following pattern for a given N.

N = 4

1

2 3

4 5 6

7 8 9 10

1

2 3

4 5 6

7 8 9 10

```
read N
Row = 1
Value = 1

while Row <= N do
    Col = 1
    while Col <= Row do
        print Value
        Value = Value + 1
        Col = Col + 1
    end

    print "\n"
    Row = Row + 1
end
exit
```



# Number Pattern - II

Write pseudocode to **print the following pattern!**

1

232

34543

4567654

567898765



# Time to Try!

- Sum of Digits of a Number
- Swap two numbers
- Half Diamond Pattern

N = 3

\*

\*\*\*

\*\*\*\*\*





# Ganesha's Pattern

Take as input N, an odd number ( $\geq 5$ ) . Print the following pattern as given below for N = 7.

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

# Lecture 2

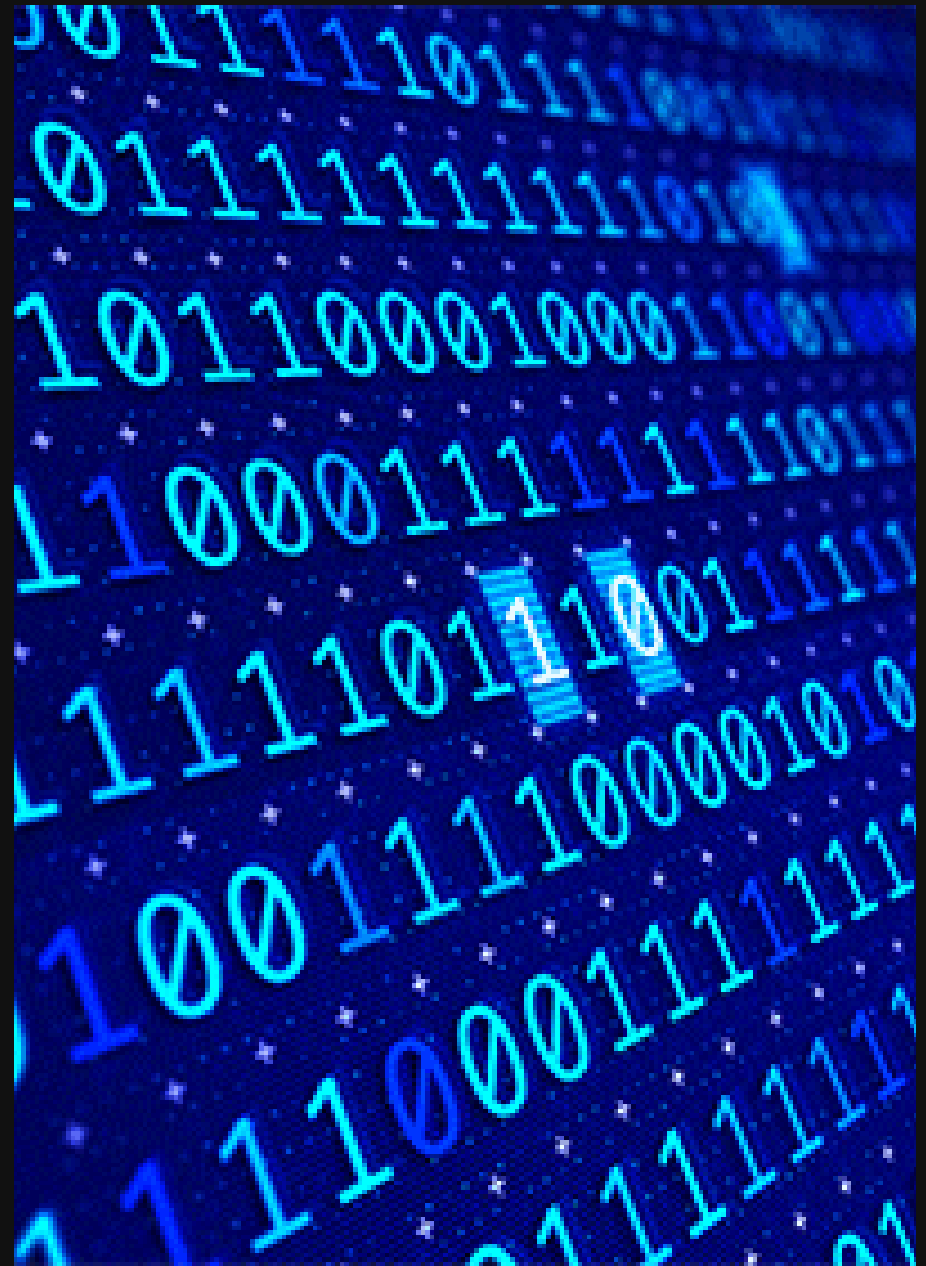
# Programming in

# JAVA





# Binary Number System



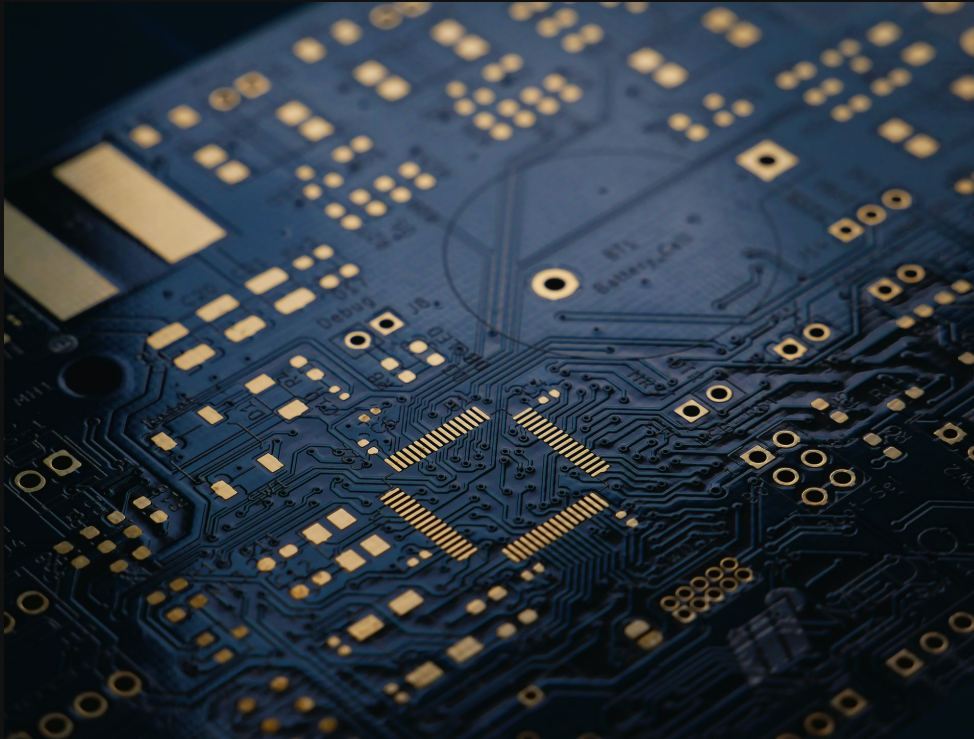
## Story behind

We humans use a decimal, or base-10, numbering system, presumably because people have 10 fingers

Early computers were designed around the decimal numbering system. This approach made the creation of computer logic capabilities unnecessarily complex and did not make efficient use of resources. (For example, 10 vacuum tubes were needed to represent one decimal digit.)



To deal with the basic electronic states of on and off, Von Neumann suggested using the binary numbering system



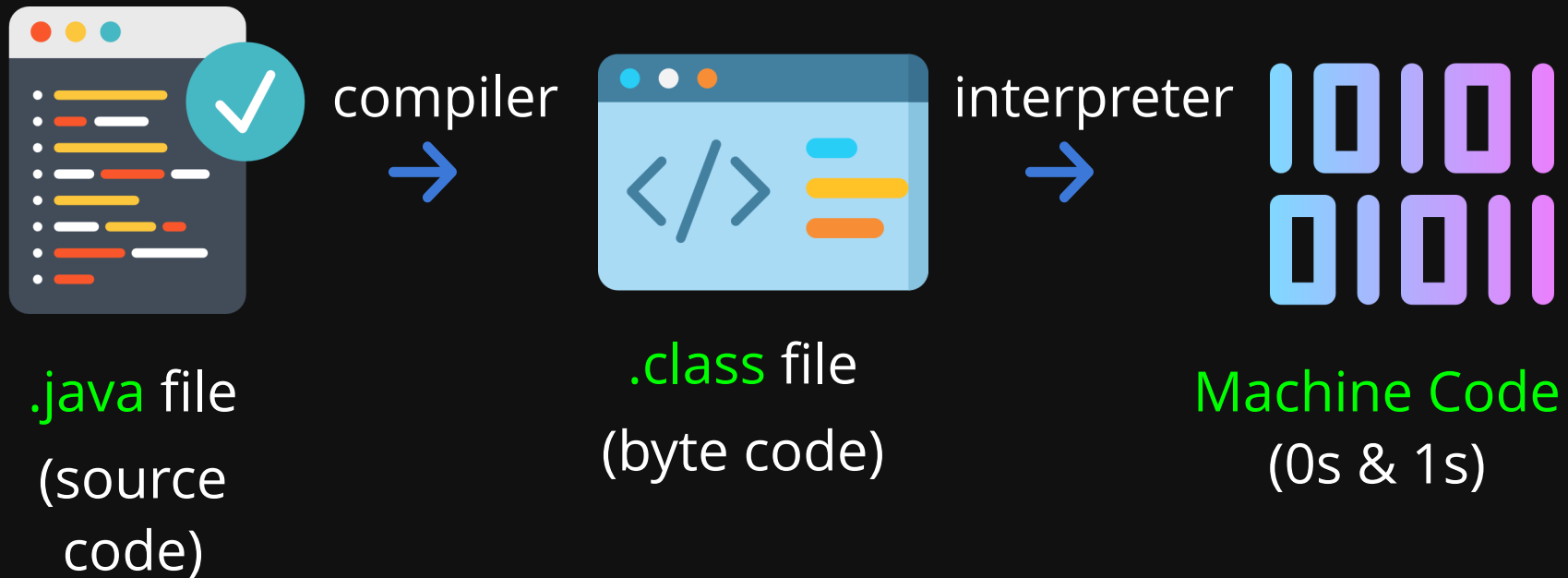
# Binary Number System



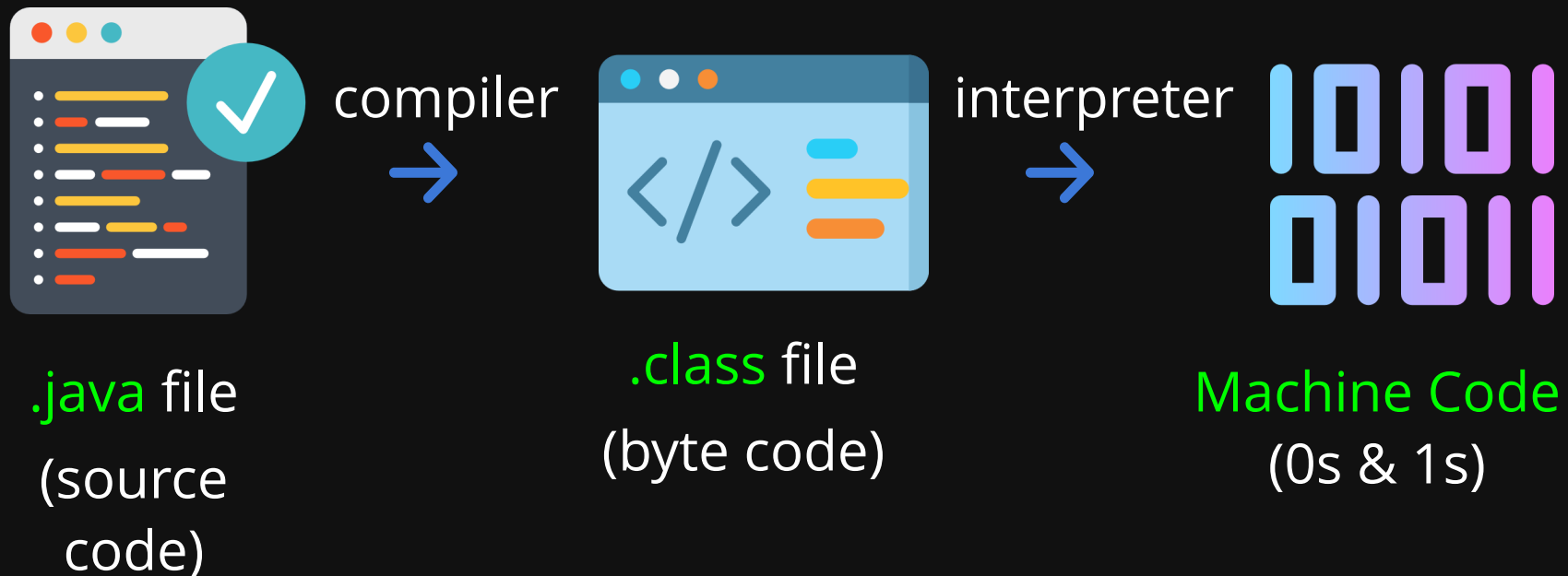


# Programming in JAVA

What happens behind the scenes?  
Why Java is Platform independent.



What happens behind the scenes?  
Why Java is Platform independent.



Byte Code doesn't run directly, we need a JVM (Java Virtual Machine) to run this code

# Platform Independence?

Byte-Code can run all  
operating systems

# C++

C/C++ Compiler generates a .exe file which is platform dependent.

# Python

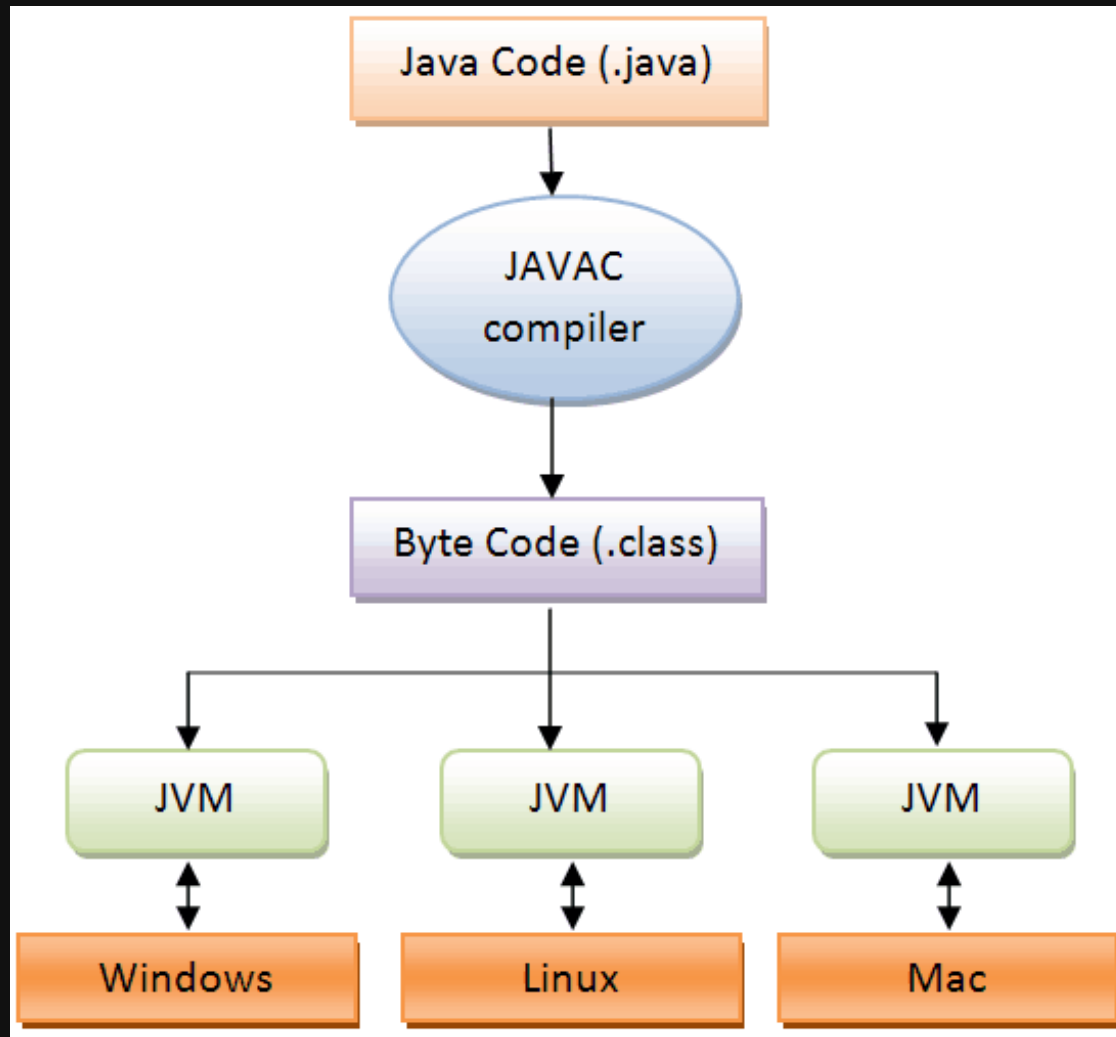
Python is interpreted language, instructions are executed line by line.

# Java

In Java we get bytecode, JVM converts this bytecode into machine code.

To run byte-code, we need a JVM installed on a machine.

Java is platform independent but JVM is platform dependent.



# Architecture

JDK = JRE + Development Tools

JRE = JVM + Library Classes

Java Virtual Machine (JVM)

JIT Compiler  
(Just In Time)



# JDK

Consists of Development tools & environment to run the Java program.

1. Development Tools
2. Compiler (javac)
3. Archiver (jar)
4. docs generator (javadoc)
5. interpreter / loader





Provides environment to only **run** the program.

JRE = JVM + Additional technologies & features

1. Deployment solutions
2. Development toolkits & libraries
3. Integration libraries
4. Language and utility libraries  
(such as collections framework)

JVM 🤔

Java Virtual Machine



# JVM

Java Virtual Machine, or JVM, loads, verifies and executes Java bytecode. It is known as the interpreter or the core of Java programming language because it executes Java code.

- **Class Loader Subsystem** is responsible for loading, linking and initializing a Java class file (i.e., "Java file"), otherwise known as dynamic class loading.
- **Runtime Data Areas** contain method areas, PC registers, stack areas and threads.
- **Execution Engine** contains an interpreter, compiler and garbage collection area.



# Summary

The JRE combines Java code created using the JDK with the necessary libraries required to run it on a JVM and then creates an instance of the JVM that executes the resulting program.

JVMs are available for multiple operating systems, and programs created with the JRE will run on all of them. In this way, the Java Runtime Environment is what enables a Java program to run in any operating system without modification.

# Software Tools [OPTIONAL] 🤔

IntelliJ Idea (IDE)

<https://www.jetbrains.com/idea/download>

(Free Community Edition)

**JDK**

<https://www.oracle.com/java/technologies/downloads/#java16>

Recommended for beginners

## Coding Minutes IDE

[ide.codingminutes.com](https://ide.codingminutes.com)



**No Installation Required!**

Directly Run | Save | Share Codes!

# First Java Program

## Boilerplate Code

```
1 //Boilerplate Code
2 class Main
3 {
4     public static void main (String[] args)
5     {
6
7     }
8 }
```

# First Java Program

## Boilerplate Code

```
1 //Boilerplate Code
2 class Main
3 {
4     public static void main (String[] args)
5     {
6         //Your Code goes here
7     }
8 }
```



# First Java Program

## Boilerplate Code

```
1 //Boilerplate Code
2 class Main
3 {
4     public static void main (String[] args)
5     {
6         //Your Code goes here
7         System.out.println("Hello World");
8     }
9 }
```

Let's Solve Problems & learn **Java syntax!**

1. Hello Java
2. Variables
3. Loops
4. Typecasting
5. Problems - Prime Numbers, Patterns, Simple Interest

