

June 10,
2015

Launchpad

Lecture - 1

Basics of Problem Solving

Anushray Gupta



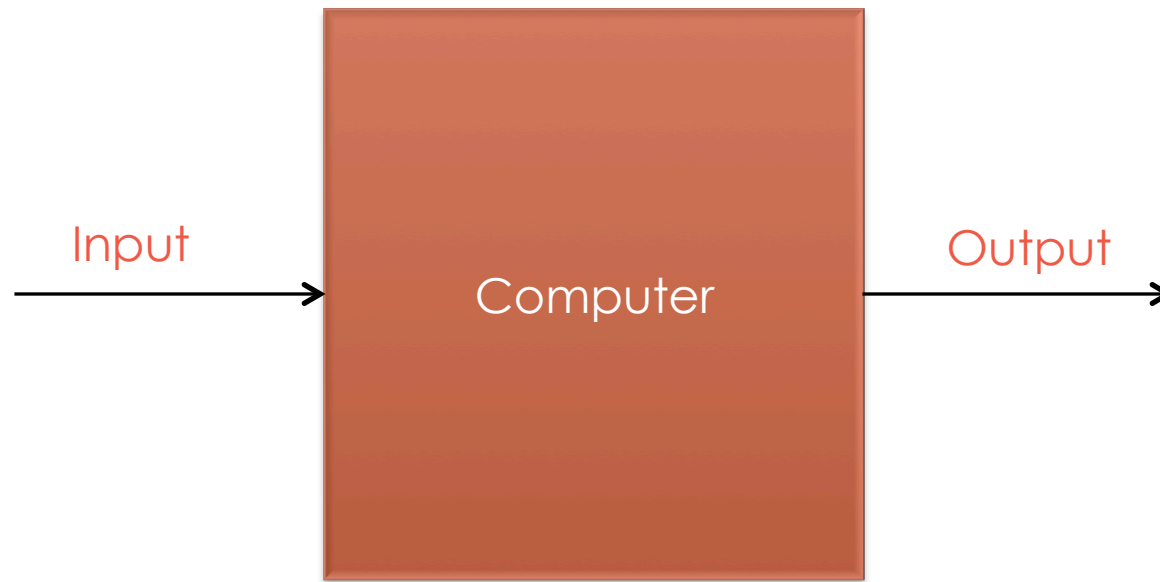
Course Structure

- I. Basics of Problem Solving
- II. Programming Fundamentals
- III. Object Oriented Programming
- IV. Data Structures

Course Administration

- I. Facebook Group
<https://www.facebook.com/groups/launchpad.batch.7/>
- II. Regular Assignments
- III. Laptops
- IV. Doubt Sessions

What does a computer do?

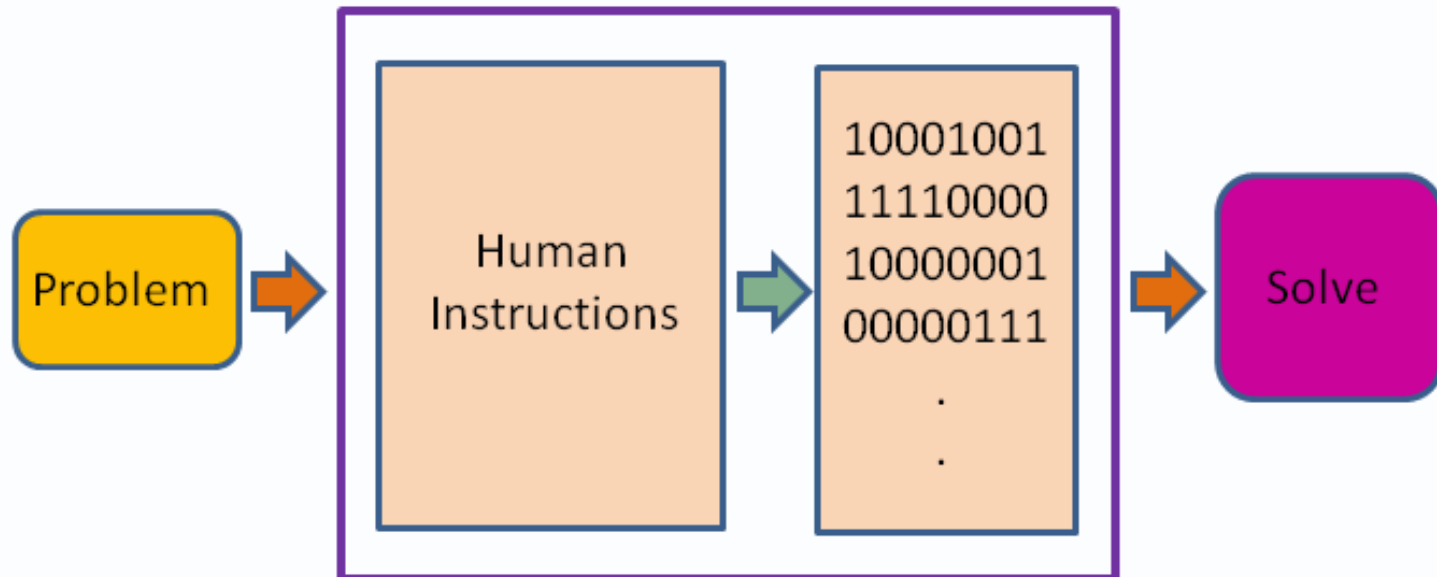


Tool for solving problems with data

- I. To communicate the solution we create **Programs**.
- II. So a program is a **sequence of instructions** that tells a computer how to perform a task.
- III. When computer follows the instructions we say it **executes** the program.

It's a machine!

- I. Computers are a machine, and at the most basic level, they are collection of switches – where **1** represents “**on**” and **0** represents “**off**”.
- II. Everything that a computer does is implemented in this most basic of all numbering systems – **binary**
- III. So if you want to communicate directly we need to talk to in binary



Programming language
(Basic , VB , C , C++ , C# , Java , Perl , ...)

What does machine understands?

Binary Instructions

I. R Type

[op | rs | rt | rd | shamt | funct]

II. I Type

[op | rs | rt | address/immediate]

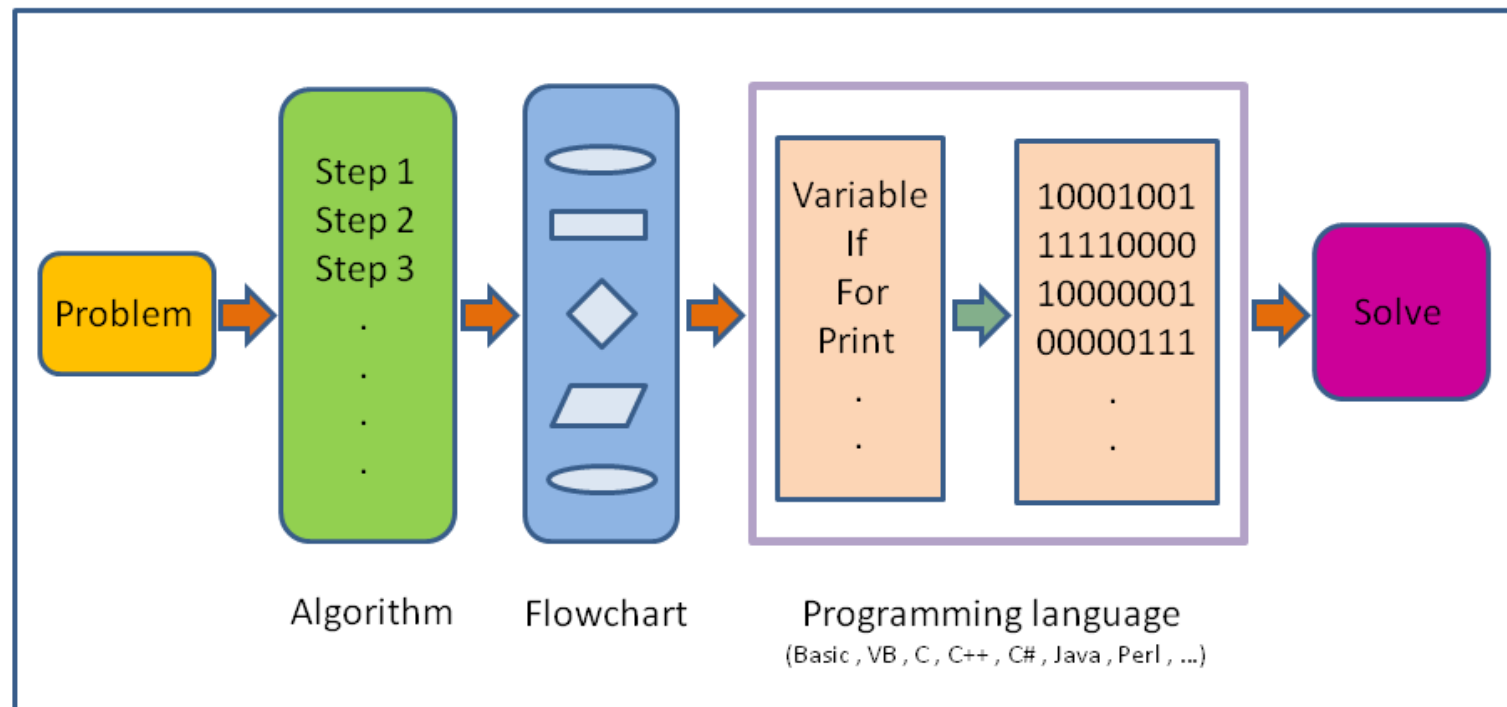
III. J Type

[op | target address]

Its nearly impossible to write in
Binary!



So we use programming language
with Flowcharts & Algorithms for
solving a Problem



What is programming language?

- I. A programming language is a set of rules that provides a way of telling a computer what operations to perform
- II. It provides a linguistic framework for describing computations.
- III. A programming language also has words, symbols and rules of grammar.
- IV. The grammatical rules are called **syntax**

High/Low Level Languages!

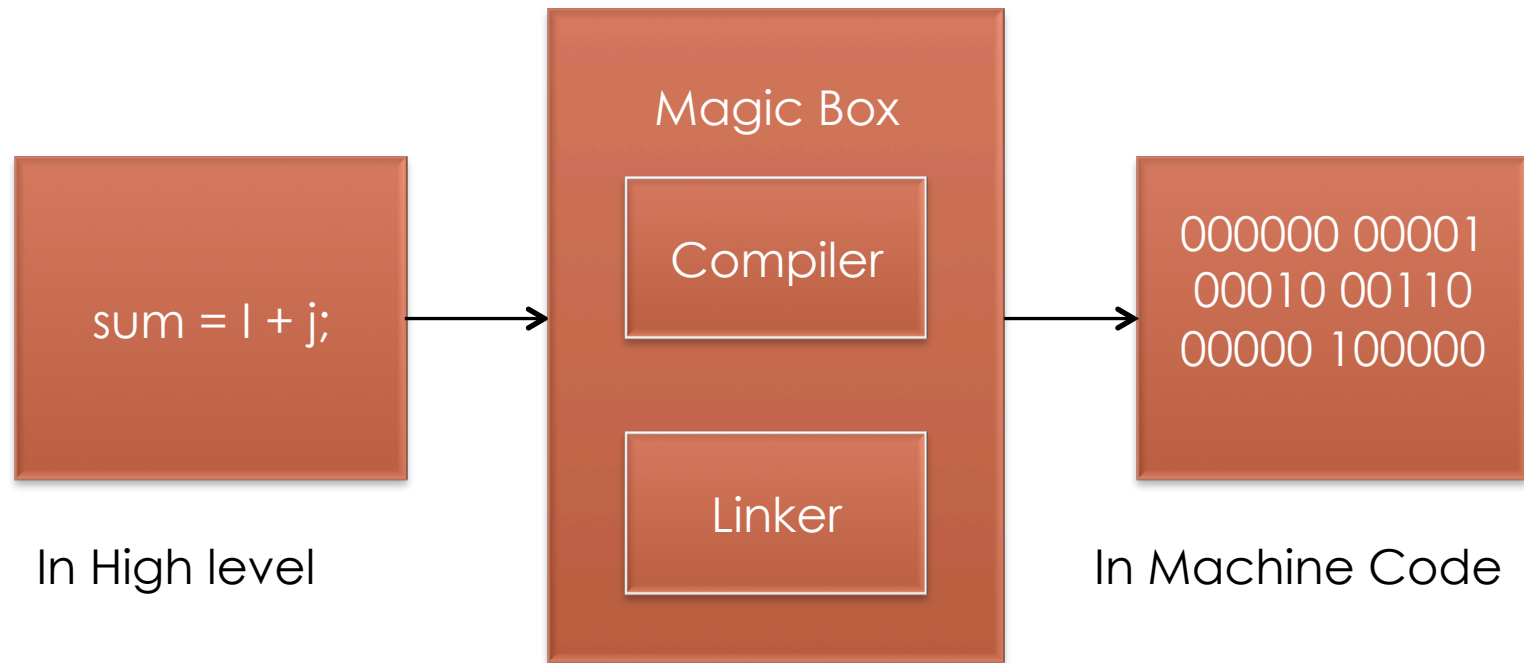
C++/ Java/ Python

Assembly language

Machine Code

ARM/MIPS/IBM

How do we work with High Level?



Before we write a program for a solution we need an **Algorithm**.



So what is an algorithm?

- I. An algorithm is a self-contained **step by step** set of operations to be performed in order to solve a problem.
- II. Its an effective method that can be expressed within **a finite amount of space** and **time** and in well-defined formal language for solving a problem.
- III. Another way to describe an algorithm is a sequence of **unambiguous** instructions.

Expressing Algorithms?

- I. Algorithms can be expressed in many kind of notations, including **natural languages**, **pseudocode**, **flowcharts**, etc.
- II. Natural Language expressions of algorithms tend to be verbose and ambiguous, and are rarely used for complex or technical algorithms.
- III. Programming languages are primarily intended for expressing algorithms in a form that can be executed by a computer.

Two basic aspects of programming

- I. Data
- II. Instructions

To understand data we need to
understand **Variables!**

What are Variables?

- I. Variables in a computer program are analogous to **Buckets** or **Envelopes** where information can be maintained and referenced.
- II. On the outside of the bucket is a name.
- III. When referring to the bucket, we use the name of the bucket, not the data stored in the bucket.

Variable Actions!

- I. **Create** one (with a nice name). A variable should be named to represent all possible values that it might contain. Some examples are: `midterm_score`, `midterm_scores`, `data_points`, `course_name`, etc.
- II. **Put** some information into it (destroying whatever was there before). We "put" information into a variable using the assignment operator, e.g., `midterm_score = 93`;
- III. **Get** a copy of the information out of it (leaving a copy inside). We "get" the information out by simply writing the name of the variable, the computer does the rest for us, e.g., `average = (grade_1 + grade_2) / 2`.

Now lets talk about Instructions!



What are instructions?

- I. Its an order given to computer.
- II. At lowest level each of these is a sequence of 0s and 1s.
- III. In assembly language, each statement is one instruction but in high level each statement can be further broken into multiple steps.

Six basic computer instructions

- I. Reading/Receiving some information
- II. Outputting/Printing some information
- III. Performing arithmetic operation
- IV. Assigning a value to a variable or memory location
- V. Conditional Execution
- VI. Repeat a group of actions

Time for Flowcharts!



What is a flowchart?

- I. Diagrammatic representation illustrating a solution to a given problem.
- II. Allows you to break down **any process into smaller steps** and display them in a visually pleasing way
- III. It shows steps as **boxes** of various kinds, and their order by connecting them with **arrows**.
- IV. It helps your audience to see the **logical flow and relationship between steps**.

Flowchart components



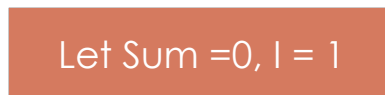
Start

Initializer / Terminator



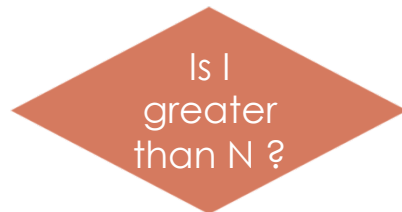
Read N

Input / Output



Let Sum = 0, I = 1

Process



Is I
greater
than N ?

Decision



Arrow

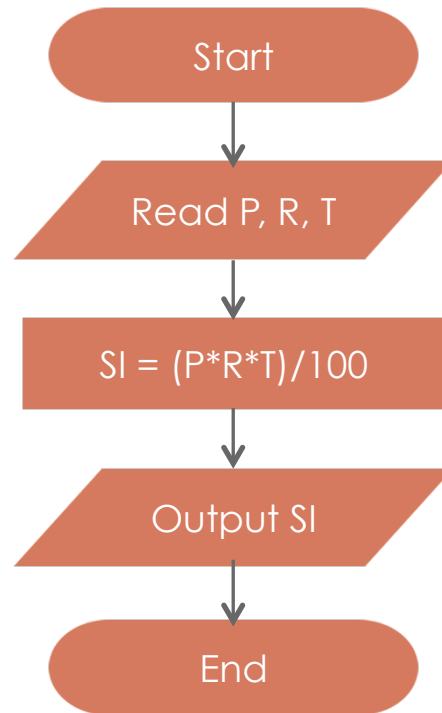


A

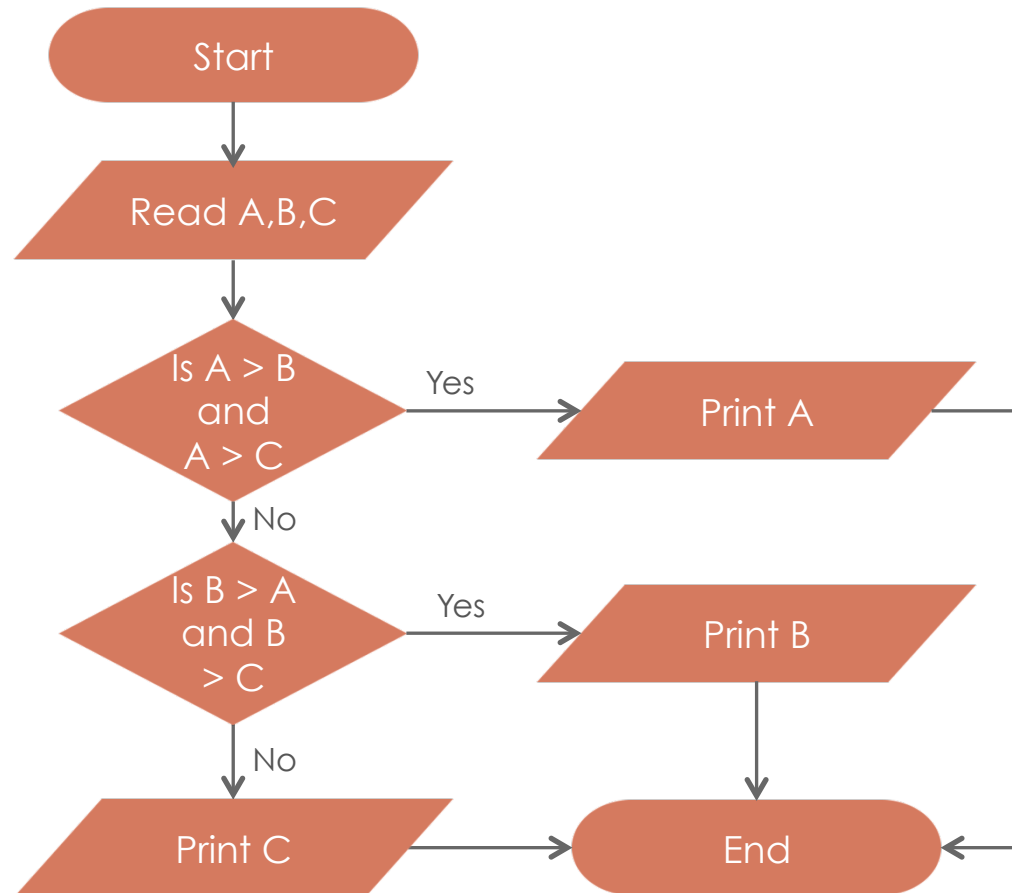
Connector

Lets look at few problems and their
flowcharts!

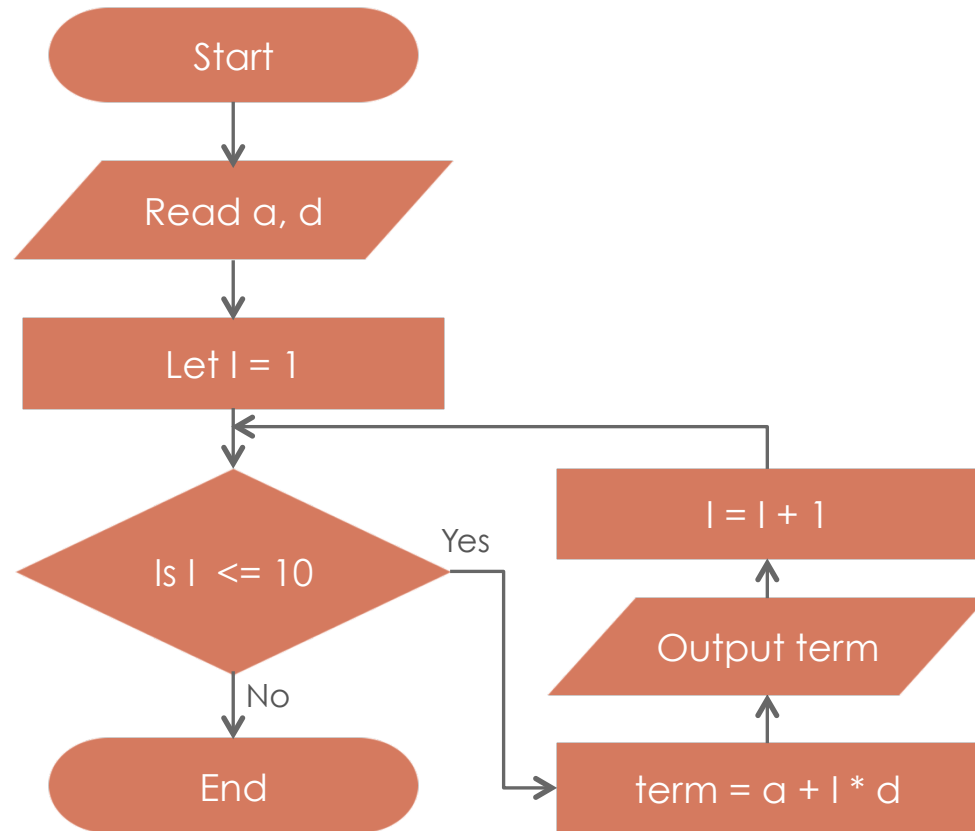
Read Principal, Rate & Time and Print SI



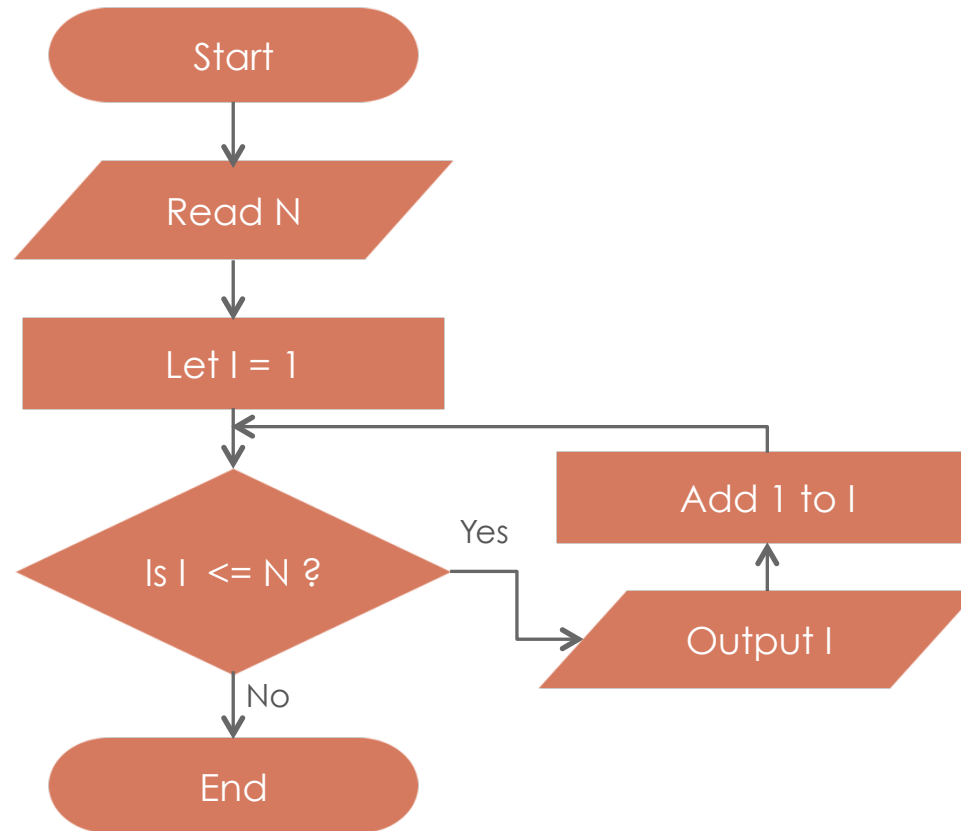
Find largest of three numbers



Read a & d , print 10 numbers of form $a+d$, $a+2d$, $a+3d$...



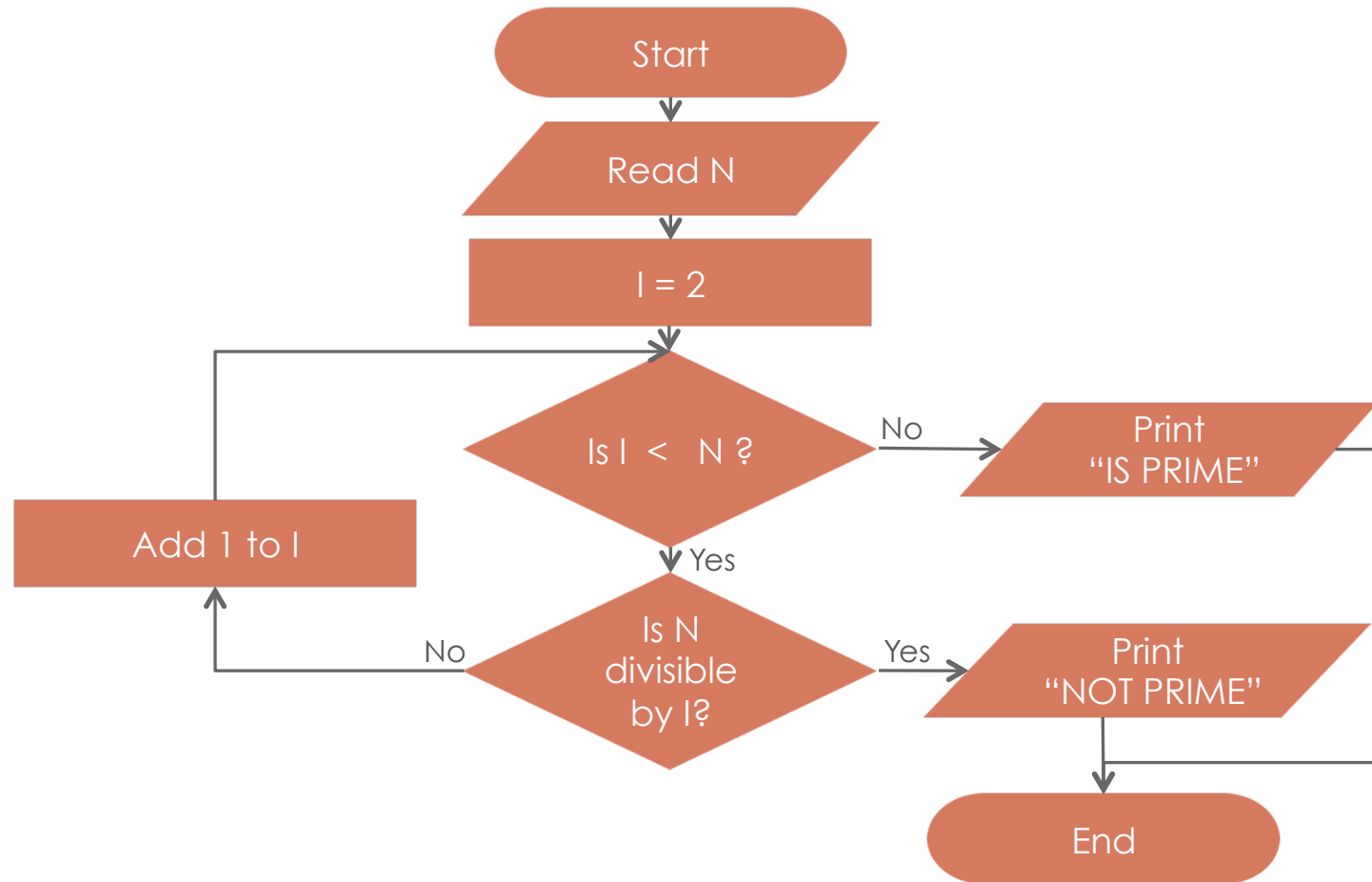
Given N, print all numbers from 1 to N



Time to try?

- I. Read five numbers and print their average
- II. Given N, find sum of even numbers from 1 to N
- III. Given N, check if its prime or not

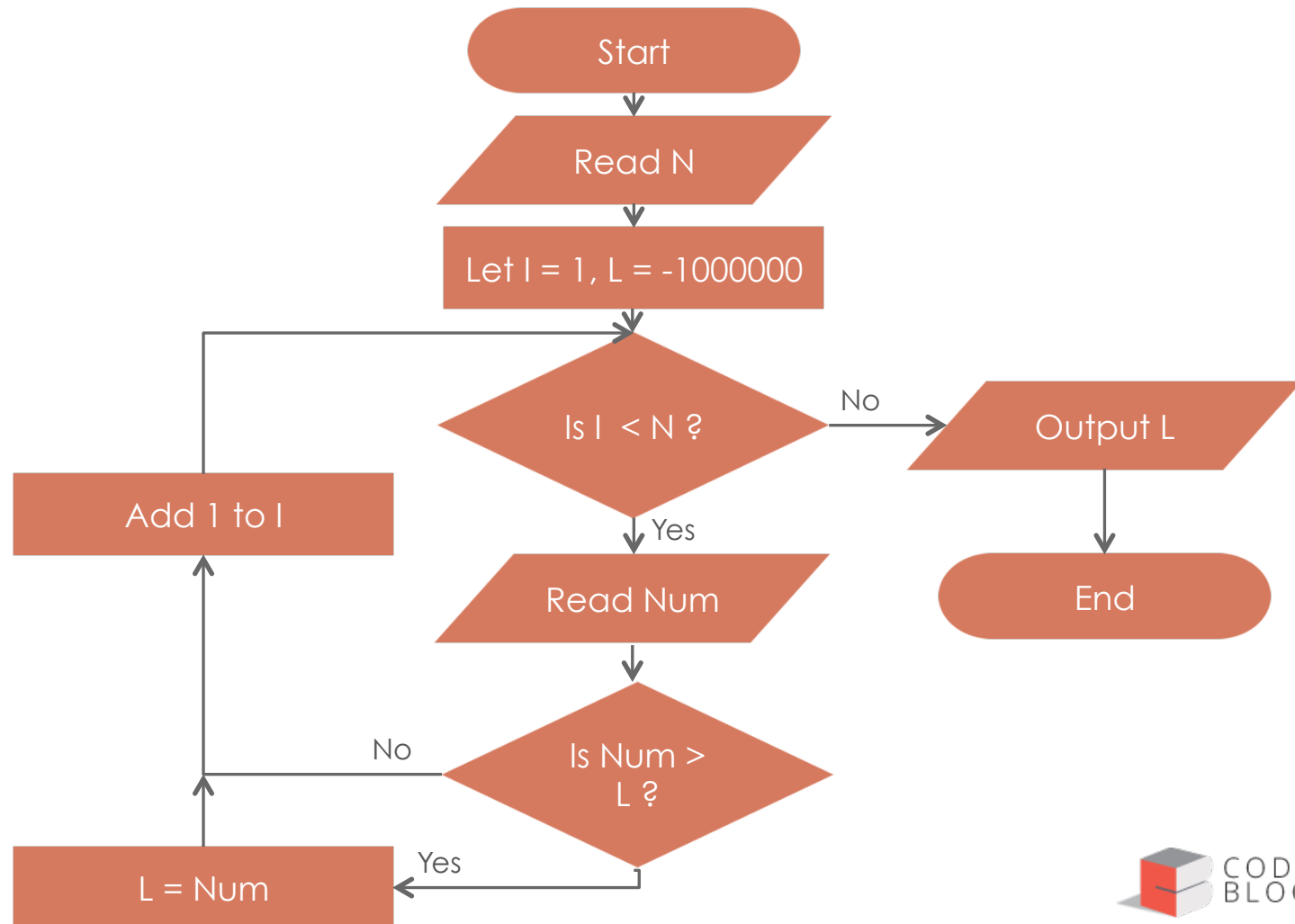
Check if a number is prime or not?



Some more examples!

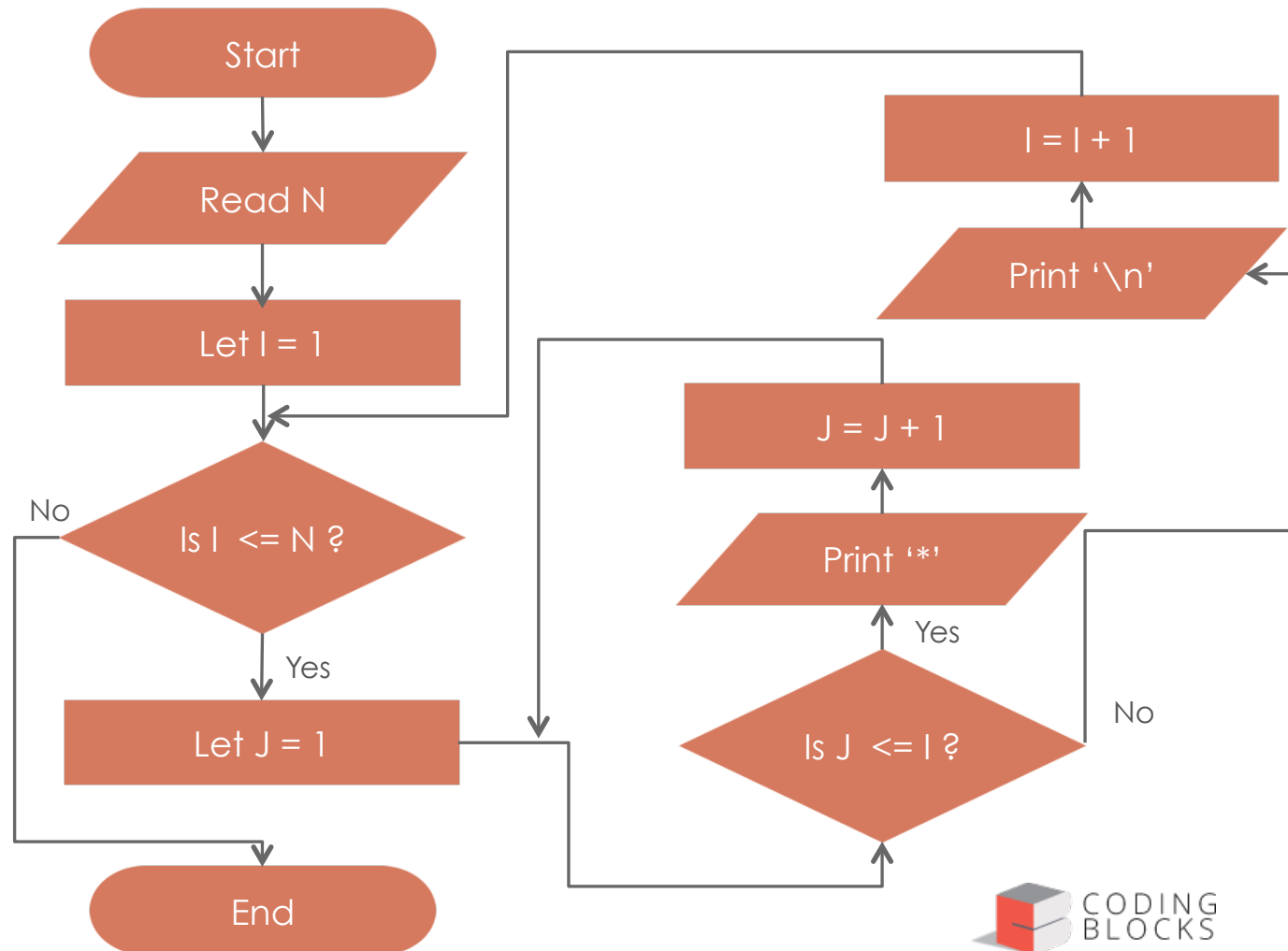


Find largest of N numbers



Print the below pattern

*
**



Time to try?

- I. Given a list of N integers, find mean, maximum and minimum value. You would be given first N, and then N integers of the list.
- II. Given a number check if it is a member of Fibonacci sequence or not
- III. Read N, and print the following pattern

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

Pseudocode!

What is pseudocode?

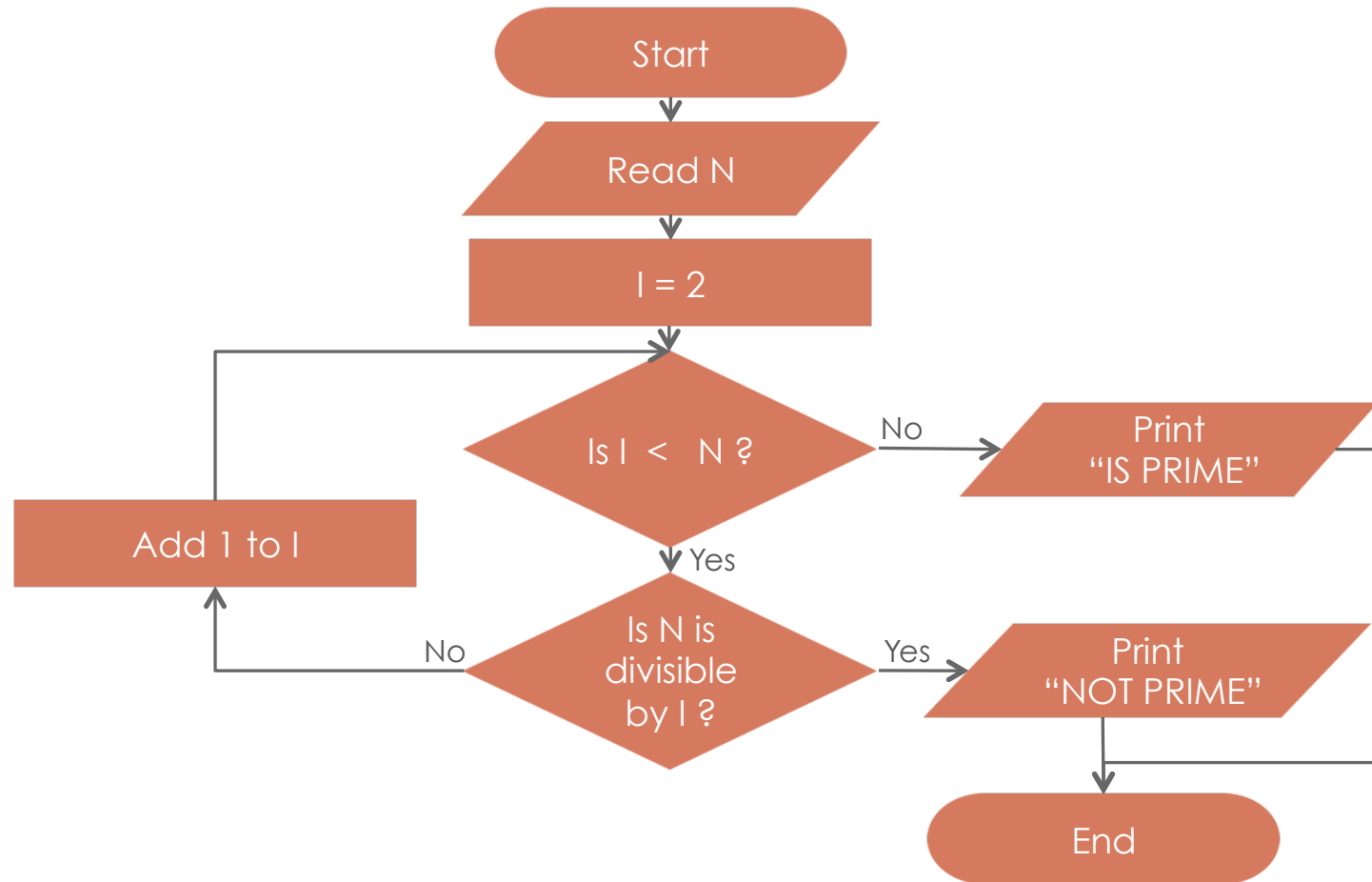
- I. A notation resembling a simplified programming language, used in program design
- II. It allows designers or lead programmers to express the design in great detail and provides programmers a detailed template for the next step of writing code in a specific programming language

Notation for those basic six!

- I. Reading/Receiving [read N]
- II. Outputting/Printing [print Sum, print "Coding Blocks", print 1, print '\n']
- III. Assignment [Sum \leftarrow 5]
- IV. Arithmetic operators [a+b, a * 5, sum + i]
- V. If Else [if I < N then ... end else then ... end]
- VI. While Loop [while I < N do ... end]

Lets convert some flowcharts into
pseudocode!

Check if a number is prime or not?



Pseudocode- Check if N is prime?

read N

$I \leftarrow 2$

While $I < N$ do

 if N is divisible by I then

 print "NOT PRIME"

 exit

 end

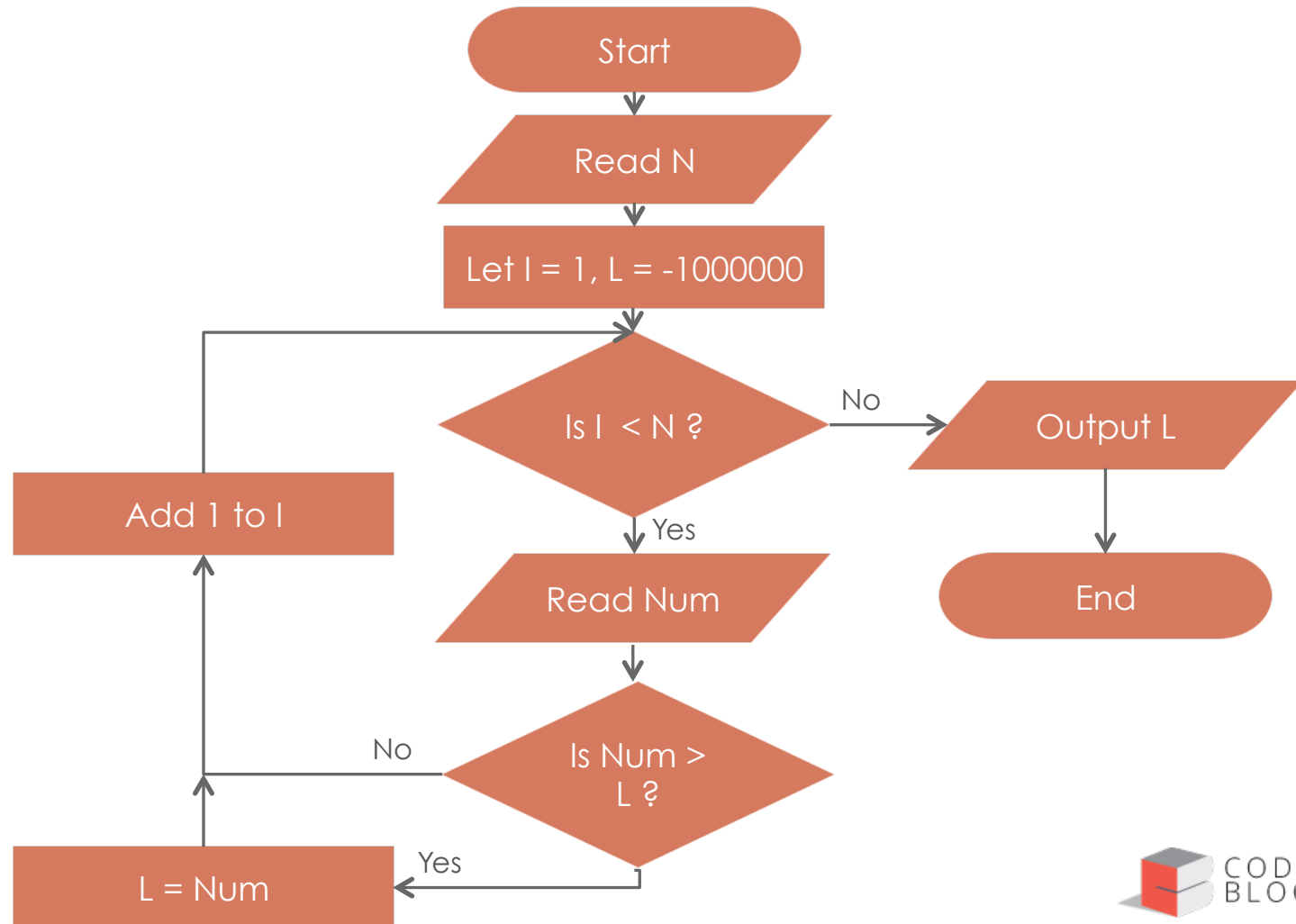
$I \leftarrow I + 1$

end

print "IS PRIME"

exit

Find largest of N numbers



Pseudocode –Largest of N numbers

```
read N
I ← 1, L ← -1000000
while I < N do
    read num
    if num > L then
        L ← num
    endif
    I ← I + 1
end
print L
exit
```

Print following pattern

1
2 3
4 5 6
7 8 9 10

```
read N
row ← 0
value ← 1
While row < N do
    col ← 0
    while col <= row do
        print value
        value ← value + 1
        col ← col + 1
    end
    print "\n"
    row ← row + 1
end
exit
```

Lets try one more pattern!

```

1
123
12345
1234567
123456789

```

```

read N
i ← 0
while i < N do
    j ← 0
    while j < n-i-1 do
        print ' '
        j ← j + 1
    end
    j ← 0, value ← 1
    while j < 2*i + 1 do
        print value
        j ← j + 1
        value ← value + 1
    end
    print '\n'
    i ← i + 1
end

```


Time to try?

- I. Convert your flowcharts into pseudocode
- II. Read N and print the below pattern

1

232

34543

4567654

567898765

- III. Given a number N, find sum of its digits

Time for Brain Teasers!



BT - 1: Hour Glasses

You have two hourglasses: a 7 minute one and a 11 minute one. Using just two hourglass, **accurately time 15 minutes.**

BT – 2: Apples and Oranges

There are three closed and opaque cardboard boxes. One is labeled "APPLES", another is labeled "ORANGES", and the last is labeled "APPLES AND ORANGES". You know that the labels are currently misarranged, such that no box is correctly labeled. You would like to correctly rearrange these labels. To accomplish this, you may draw only one fruit from one of the boxes. **Which box do you choose, and how do you then proceed to rearrange the labels?**

BT – 3: Average Salary

Three coworkers would like to know their average salary. However, they are self-conscious and don't want to tell each other their own salaries, for fear of either being ridiculed or getting their houses robbed. **How can they find their average salary, without disclosing their own salaries?**

BT – 4: Criminal Cupbearers

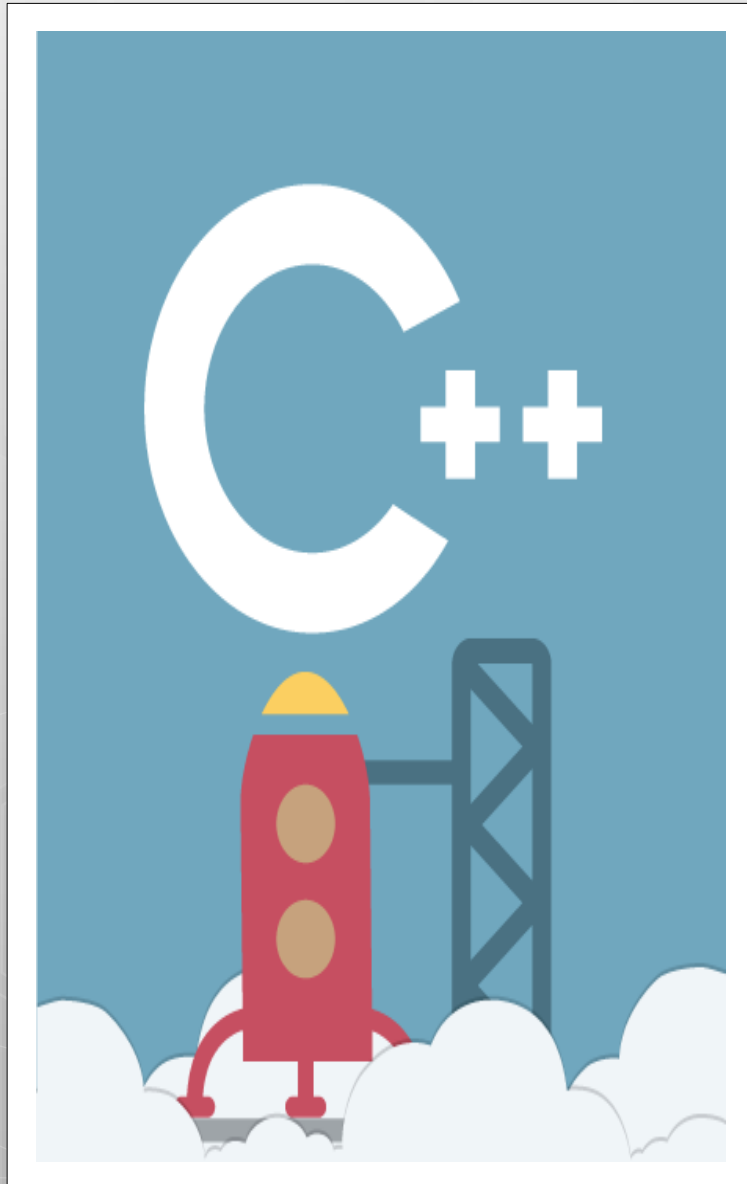
An evil king has 1000 bottles of wine. A neighboring queen plots to kill the bad king, and sends a servant to poison the wine. The king's guards catch the servant after he has only poisoned one bottle. The guards don't know which bottle was poisoned, but they do know that the poison is so potent that even if it was diluted 1,000,000 times, it would still be fatal. Furthermore, the effects of the poison take one month to surface. The king decides he will get some of his prisoners in his vast dungeons to drink the wine. **Rather than using 1000 prisoners each assigned to a particular bottle, this king knows that he needs to murder no more than 10 prisoners to figure out what bottle is poisoned, and will still be able to drink the rest of the wine in 5 weeks time. How does he pull this off?**

What is next class about?



Programming Fundamentals - 1

- I. Basic syntax of C++
- II. Datatypes/Variables
- III. Constants
- IV. If else
- V. while



Thank You!

Anushray Gupta

anushray@codingblocks.com
+91-9555567876
