

Coding ? Programming ?

Introduction

Welcome in the world of Coding!

Course objective

- Building blocks
- Logical & Critical Thinking Skill
- Learn a programming language

Modules

- Flowchart
- Pseudocode
- Coding,

- Who can join? Anyone / Everyone
- ✓ computer science / Non-computer science
 - ✓ Engineering / School / College Students
 - ✓ Working Professionals.

Let's start with a simple problem statement

Problem: Make a cup of Tea

Solution: You will come up with a list of steps that you will take to make that cup of tea.

So, the sequence of steps, we call it as an algorithm

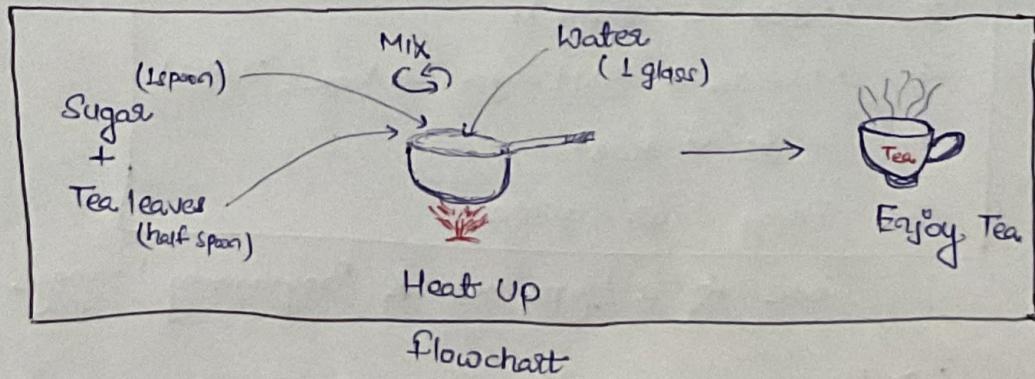


Algorithm is the step by step solution to a particular problem.

So, Algorithm is a sequence of instructions to solve a particular problem.

Algorithm can be depicted using flowchart or pseudocode or real code.

Let's see the flowchart to make Tea :-



Flowchart is a diagrammatic representation of the process or steps that you are taking to solve a particular problem.

Pseudocode.

1. Take 1 spoon Sugar and 1/2 spoon Tea leaves
2. Take 1 glass of water
3. Mix Sugar & Tea leaves in water
4. Heat for 2-3 minutes
5. Enjoy your Tea

You can convert your flowchart into a code that is written in plain English.

Now, If you want 5 cups of Tea, then you will have to increment the amount of Sugar, Tea and water.

* You have to be very specific to the constraints of the problem. If someone ask you make ~~some tea~~?

You have to define what is some?

In Programming, you can't write ambiguous instructions.

Because Compiler or interpreter that needs to be clear.
What exactly is the amount that to consider?

किसी फूलना क्या है? X
कहों से किसी फूलना क्या है? ✓

Next you can convert your pseudocode into some real code.

1. Sugar = input()
2. Tea leaves = input()
3. Water = input()
4. Mixture = Add(Sugar, Tea leaves, Water)
5. Heat(Mixture)
6. Enjoy your coffee

Code in a Hypothetical Programming Language

→ You can write this code in any programming language like C or C++, Java, Python, Go, Rust, Javascript etc.

| You don't understand. Don't worry.

We are here to break down the fundamental programming concept into very simpler terms.

Let's Begin

Flowcharts

Building Blocks of Flowcharts

Flowcharts?

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

So, It become very easy for beginner to draw and understand a flowchart without writing complex code, you will easily understand what steps you will perform in order to solve a problem.

Flowchart Components

1. First building block / component of flowchart is—

Start

It denotes the starting step that means the solution starts from this particular box.

Start/ End

End

It can also be used to terminate the process. If you want to end the process you can draw a box like this.

2. Reading data or Read the input and show the output

Read N

Input/ Output

If you want to make tea, you will read the input for how many guests you want to make? So, you will the value of N, that will denote the number of guests who want to have tea.

So, we will store this value inside a bucket N, that bucket is created inside RAM.



↑ Random Access Memory

Print N

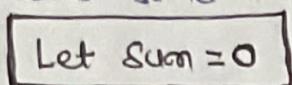
Similarly to print something, we use same box.

Print "Hi"

Print N

" " → double quote.

3. Third component is rectangle, that is used anywhere you have to process some data.



Process

e.g.: marks_{avg} = $\frac{\text{Math} + \text{Phy} + \text{Chem}}{3}$

Assignment

Sum = 0

assigning the value 0 to the bucket Sum.

So, all these instructions where processing is happening, anything except from input output they will be depicted as in rectangle

So, Now sum bucket will store Value 0.

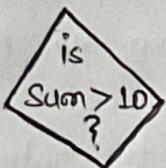
= \vee ==

Assignment

Equality

These two are different operators in programming.

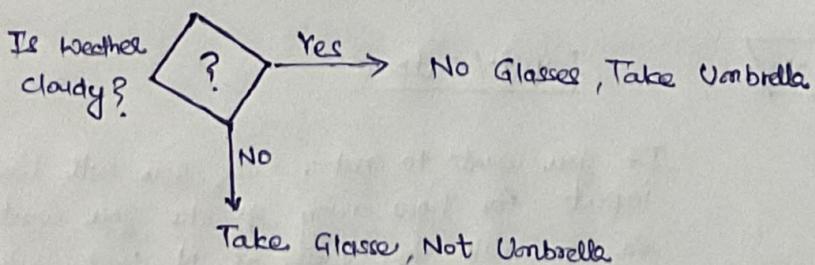
4. The 4th component is decision box, also known as rhombus



Decision

It is denoted using the diamond shape box.

Whenever you have certain condition to check which mean you want to check whether the condition is true/false based upon that there can be two branches.

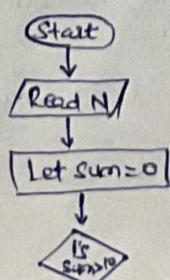


5. The fifth and last component of flowchart is Arrows.



Arrows are used to connect two different boxes.

Connection



① Flowchart - Simple Interest calculator

Read Principal Amount, Rate of Interest and Time and print Simple Interest.

Sample Input

$$P = 100$$

$$R = 5$$

$$T = 2$$

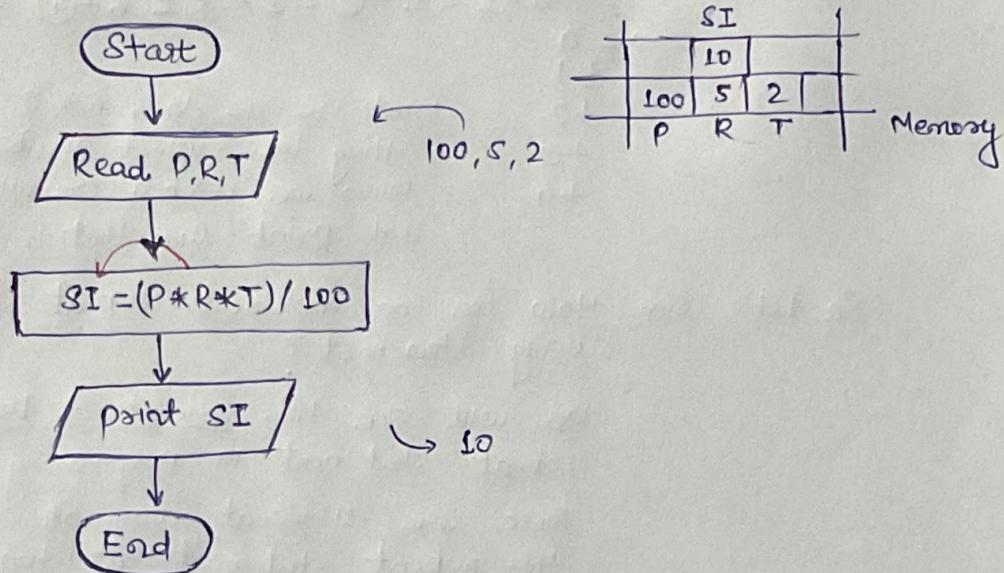
Sample output

$$SI = 10$$

$$SI = \frac{P * R * T}{100}$$

$$= \frac{100 * 5 * 2}{100}$$

$$= 10.$$



② Flowchart - Largest Number

Given 3 numbers, find the largest number.

Sample Input

$$A = 10$$

$$B = 30$$

$$C = 20$$

Sample output

$$30$$

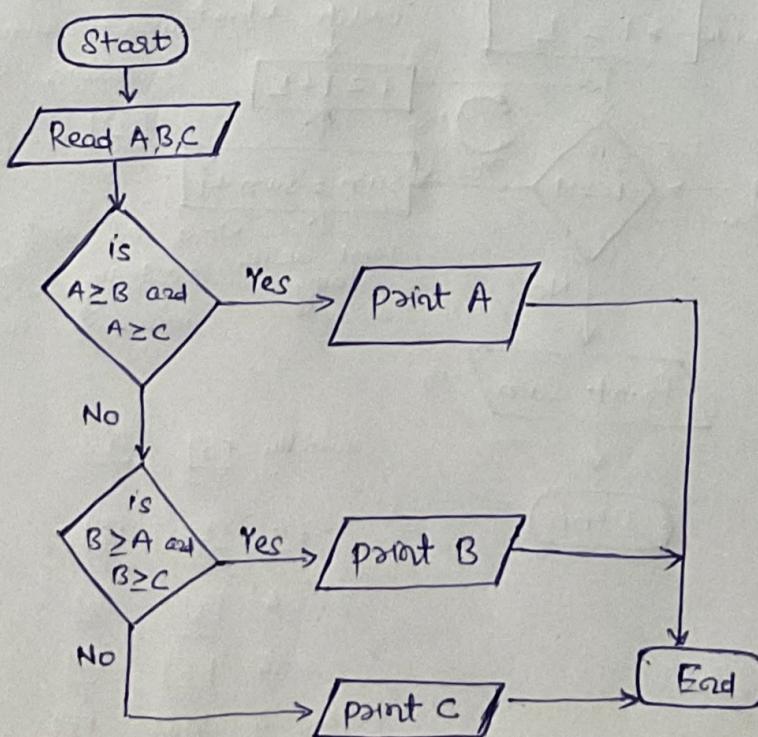
$$10, 30, 20$$

$$12, 8, 10$$

$$10 \geq 30 \text{ and } 10 \geq 20$$

$$30 \geq 20$$

why \geq ?
bcuz 10, 20, 20



⑥ Flowchart - Sum of first N numbers.

Find the sum of numbers from 1 to N,
N is given as input.

Example:- Input Output $1 + 2 + 3 + 4 = 10$
 $N = 4$ 10

Here, we can observe one thing we are adding, again and again
Repeating Sum = 0

$$0 + \textcircled{1} = 1 + \textcircled{2} = 3 + \textcircled{3} = 6 + \textcircled{4} = 10$$

$$\text{Sum} = 0$$

$$+ 1$$

$$+ 2$$

$$+ 3$$

$$+ 4$$

what we are doing here?

Adding the current number till N.

When we reach 4 we will stop
and print sum that is 10.

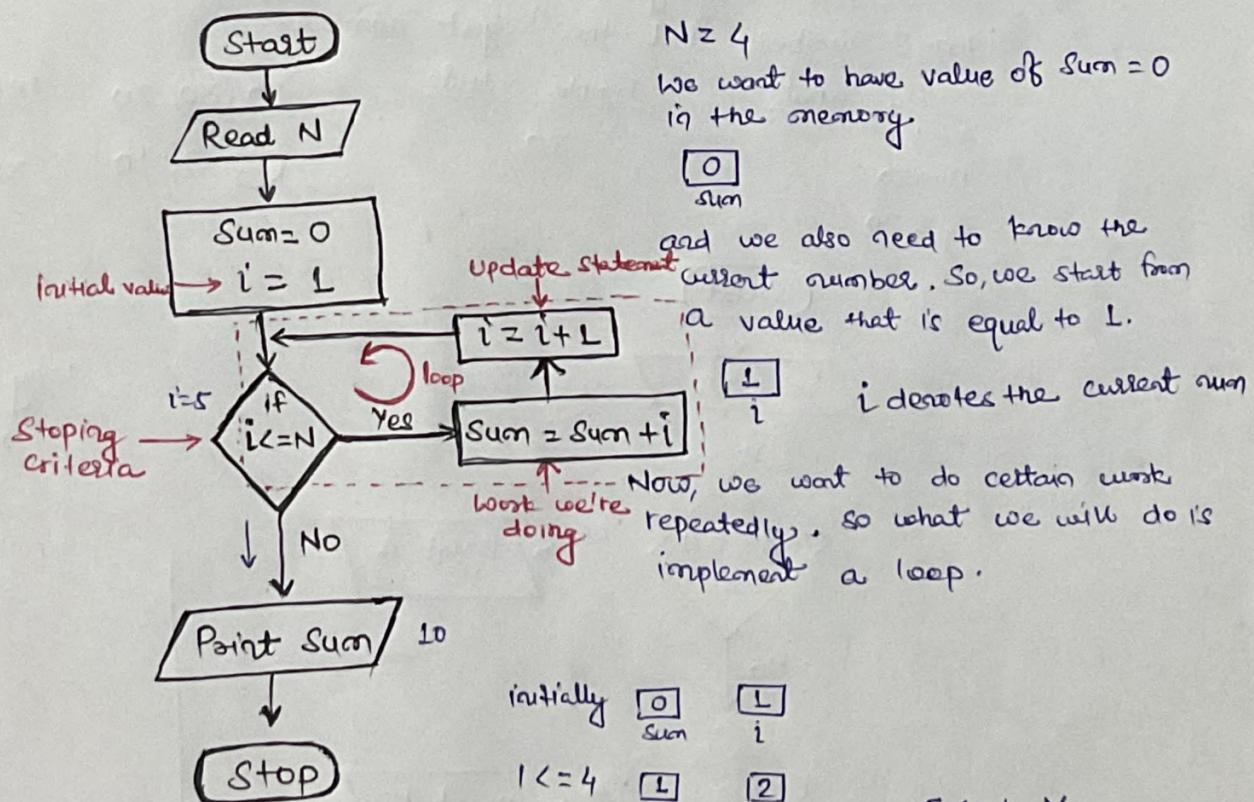
So, Let's see, How we can implement repeat statement using flowchart?

We will see the concept of loop for repeat statement in programming language.

There are different ways of implementing the repeat statement/instructions.

(i) For loop

(ii) While loop



initially	Sum	i
	$\boxed{0}$	$\boxed{1}$
$1 \leq 4$	$\boxed{1}$	$\boxed{2}$
$2 \leq 4$	$\boxed{3}$	$\boxed{3}$
$3 \leq 4$	$\boxed{6}$	$\boxed{4}$
$4 \leq 4$	$\boxed{10}$	$\boxed{5}$

$5 \leq 4 X$

④ Flowchart - Prime Numbers

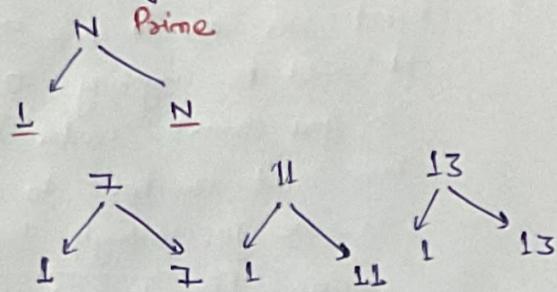
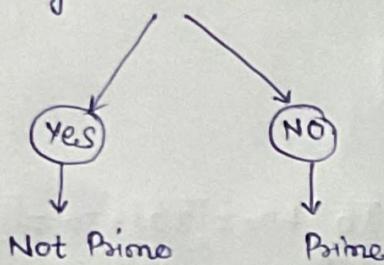
Given a Number, check if it is Prime or Not.

<u>Input</u>	<u>Output</u>
11	Yes
6	No

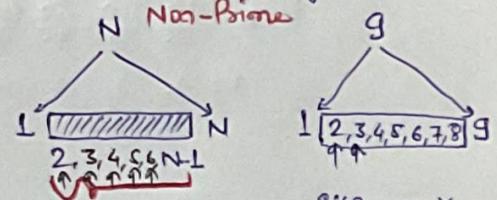
Prime numbers is a number that has only two dividers 1 and itself.

Logic :-

Find out a divisor in the range 2 to $N-1$.

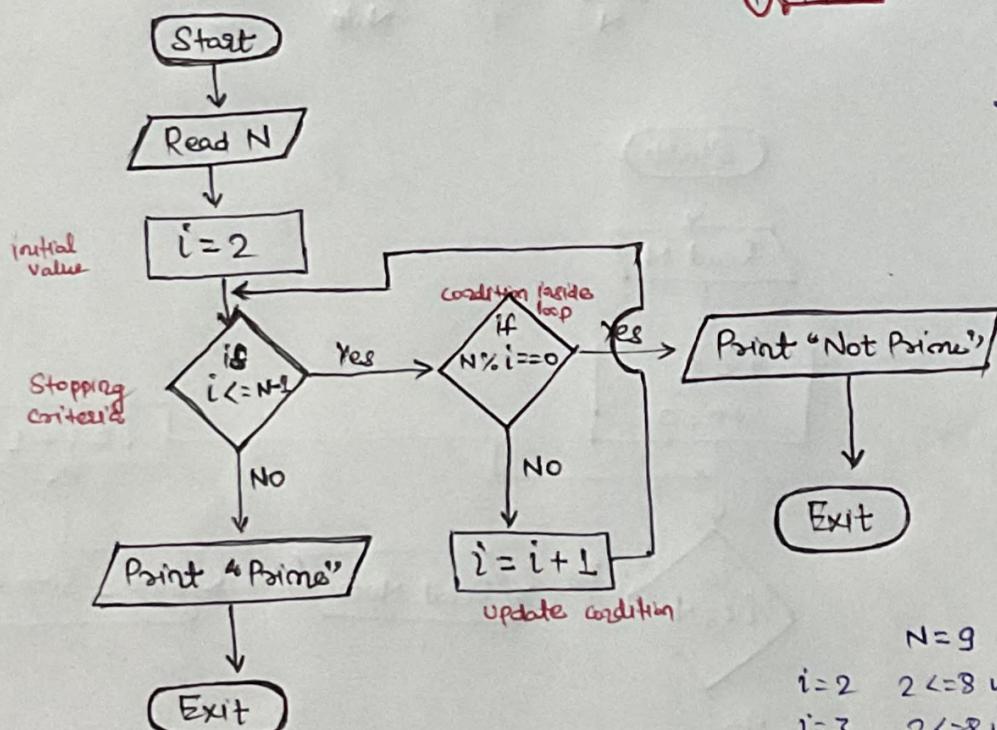


Non-Prime Numbers will have at least one divisor in the range 2 to $N-1$.



$$9 \times 2 = 20 \times$$

$\therefore 9$ is not prime



Dry run :- N = 7

$i = 2 \quad 2 <= 6 \checkmark \quad 2 \% 2 == 0 X \quad i = 2 + 1$

$i = 3 \quad 3 \leq 6 \quad \checkmark \quad 7 \% 3 == 0 \quad x \quad i = 3 + 1$

$i = 4$ $4 <= 6 \checkmark$ $7 \% 4 == 0 \times$ $i = 4 + 1$

$i = 5$ $S / = 6 \vee$ $\exists S = 8 \times i = 5 + 1$

$i = 6$ $6 \times 5 \checkmark$ $7 \times 6 = 30 \times$ $i = 6 + 1$

$i = 7$ $7 <= 6$ x
 \hookrightarrow ~~Position 4~~ exit.

Not Prime
Exit

$$\begin{array}{ll} i=2 & 2 < 8 \vee 9 \% 2 = 0 \quad x \\ i=3 & 3 < 8 \vee 9 \% 3 = 0 \quad \checkmark \end{array}$$

① Flowchart - Sum of Multiple Inputs

Take input N followed by N more numbers, find the sum of those numbers.

Example:-

N = 4

10, 20, 30, 50

Output
110

$$10 + 20 + 30 + 50 = 110$$

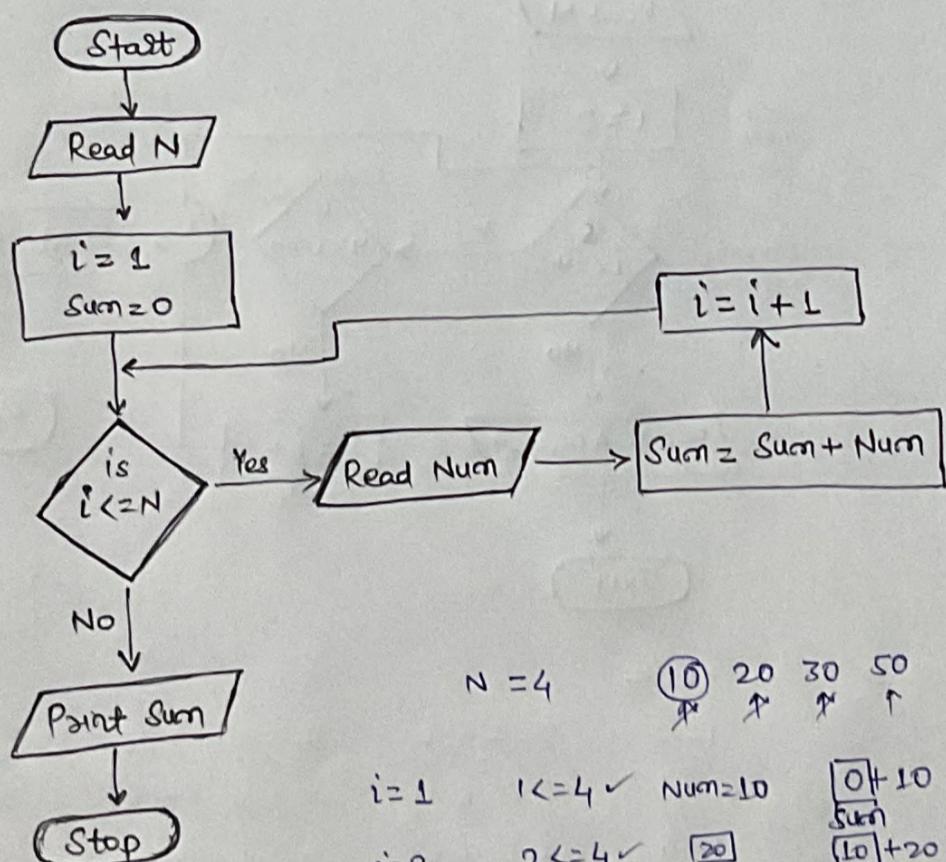
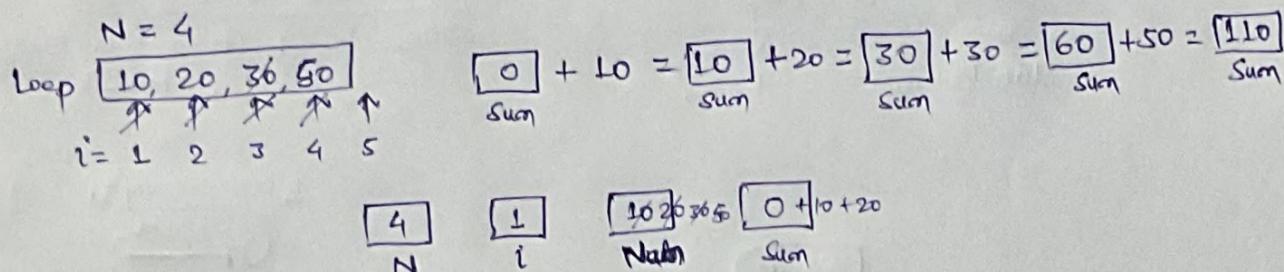
First we have to see How we can take N inputs from User.

N could be anything, 50, 100.

So, you cannot create N numbers of buckets to read.

then what we have to do? is read and process the sum as we go through the input.

means - [Read and Update sum]



$$N = 4 \quad \boxed{10} \ 20 \ 30 \ 50$$

$i = 1$	$1 \leq 4 \checkmark$	$\text{Num} = 10$	$0 + 10 = 10$	$i = 12$
$i = 2$	$2 \leq 4 \checkmark$	$\boxed{20}$ Num	$10 + 20 = 30$	$i = 23$
$i = 3$	$3 \leq 4 \checkmark$	$\boxed{30}$ Num	$30 + 30 = 60$	$i = 34$
$i = 4$	$4 \leq 4 \checkmark$	$\boxed{50}$ Num	$60 + 50 = 110$	$i = 45$
$i = 5$	$5 \leq 4 X$		$\text{Sum} = 110 \checkmark$	exit.

* Here, In this problem we do not need to store all the numbers at once. We're only storing current num. But in some problem we might require to store all numbers \rightarrow Array Data Structure.

⑥ Flowchart - Greatest Common Divisor (GCD)

Find GCD of two numbers.

$$a = 8$$

$$\gcd \rightarrow 4$$

$$a=8$$

$$\gcd \rightarrow 8$$

$$a = g$$

$\gcd \rightarrow L$

GCD or HCF

↓

What is the largest sum that divides both two numbers?

baz there is no common divisor except 1.

If you look carefully, you can say—

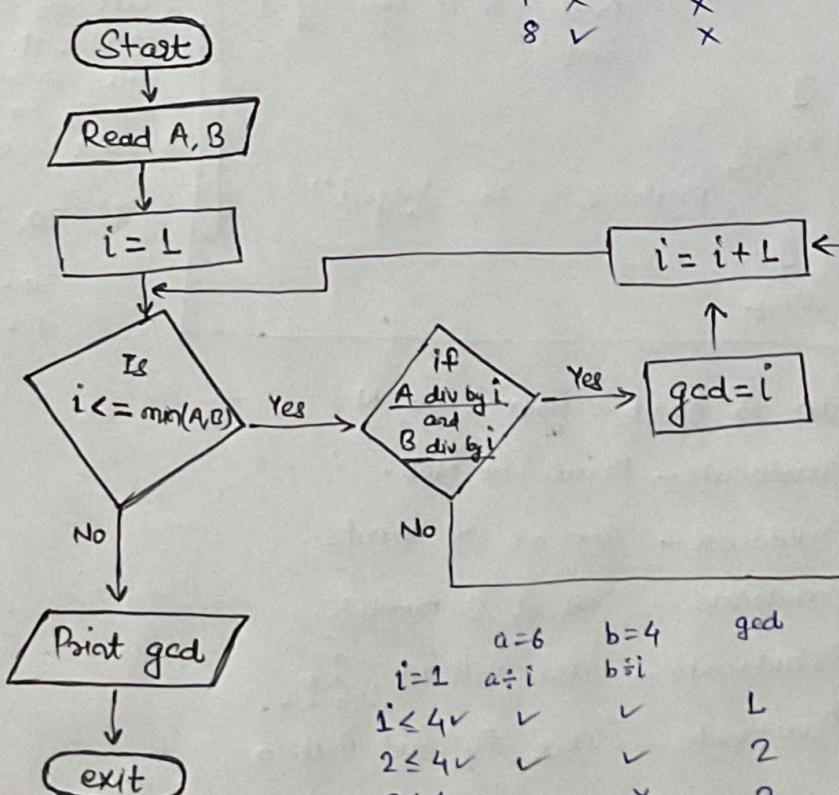
$$a, b \quad \gcd(a, b) \leq \min(a, b)$$

\therefore we can say - God range can lie in the
range L to $\min(a, b)$

Answer will lie ↑ loop.

Now we can easily iterate over this range and find out greatest number in this range which divides both a, b.

<u>Example:-</u>	$a = 8$	$b = 20$	gcd
i = 1	✓	✓	8
2	✓	✓	2
3	X	X	2
4	✓	✓	4
5	X	✓	4
6	X	X	4
7	X	X	4
8	✓	X	4



We approached it through Brute force.
There is an optimized solution for it
known as Euclid Algorithm.

	$a=6$	$b=4$	gcd	
$i=1$	$a \div i$	$b \div i$		
$i \leq 4 \vee$	✓	✓	L	$i=2$
$2 \leq 4 \vee$	✓	✓	2	$i=3$
$3 \leq 4 \vee$	✓	X	2	$i=4$
$4 \leq 4 \vee$	X	✓	[2]	$i=45$
$5 \leq 4 X$				
	L $\rightarrow \text{gcd}=2 \rightarrow \text{exit}$			

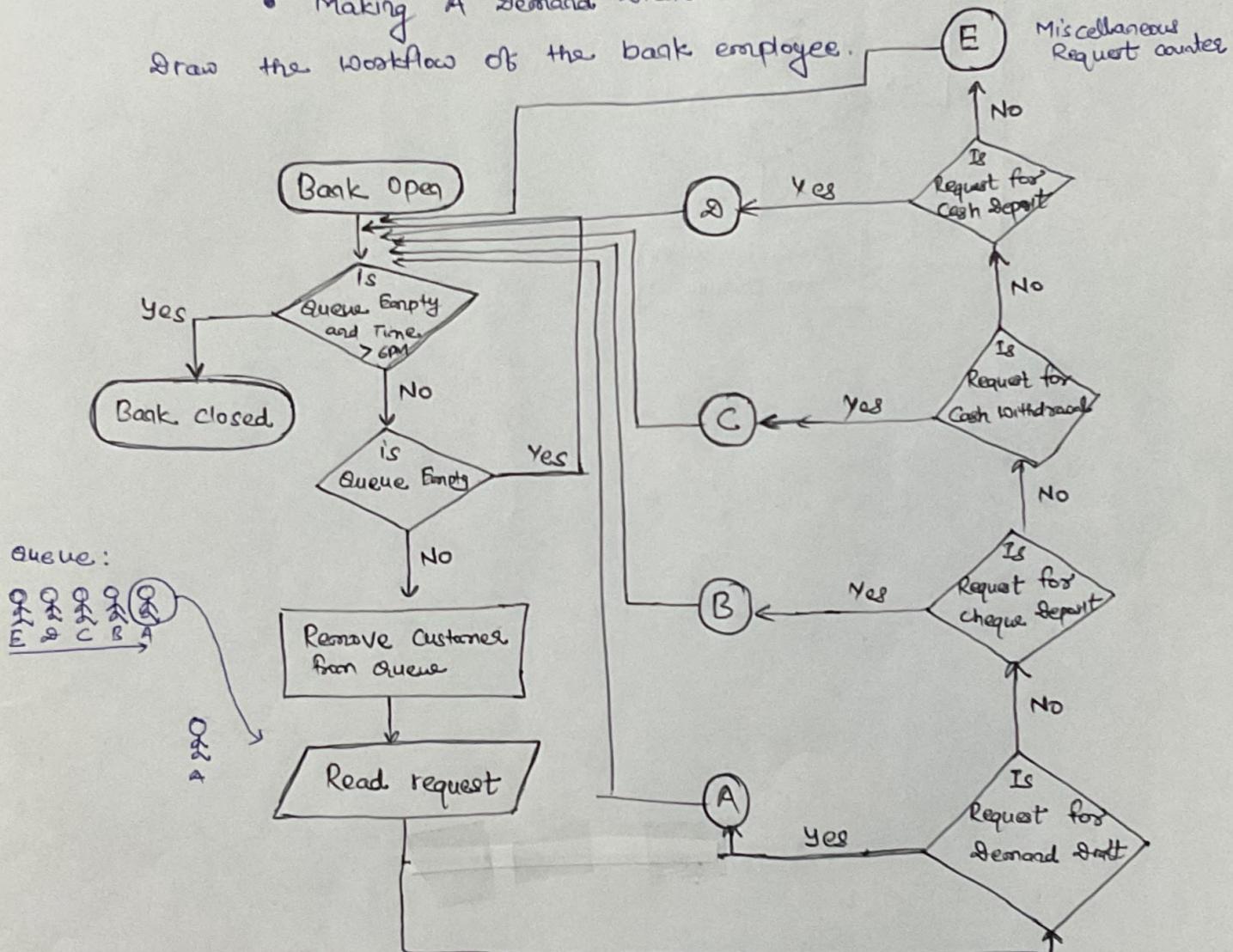
Flowchart - Bank Employee

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

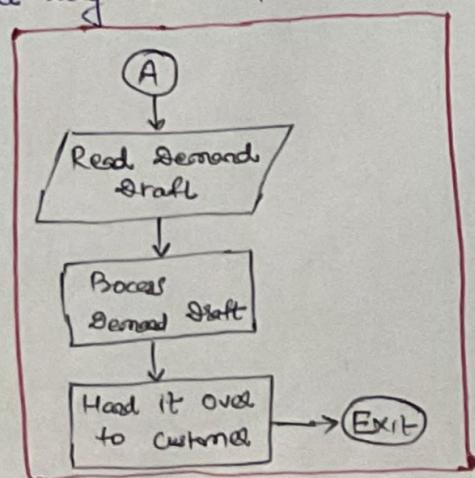
Requests can be one of the four types:-

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

Draw the workflow of the bank employee.



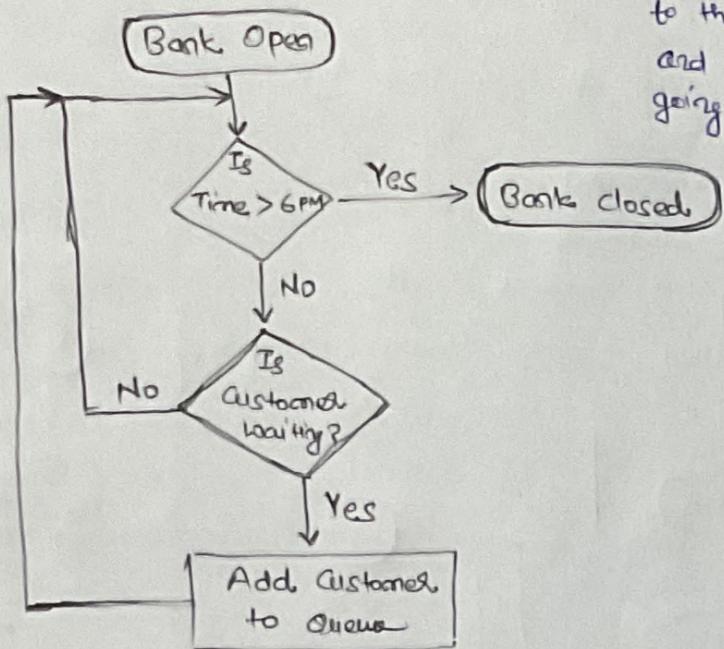
You may have separate flowchart of each ① ② ③ ④ ⑤ Actions



Connectors → They combine two different flowcharts.

Flowchart - Bank Guard

Flowchart of Bank Security Guard,



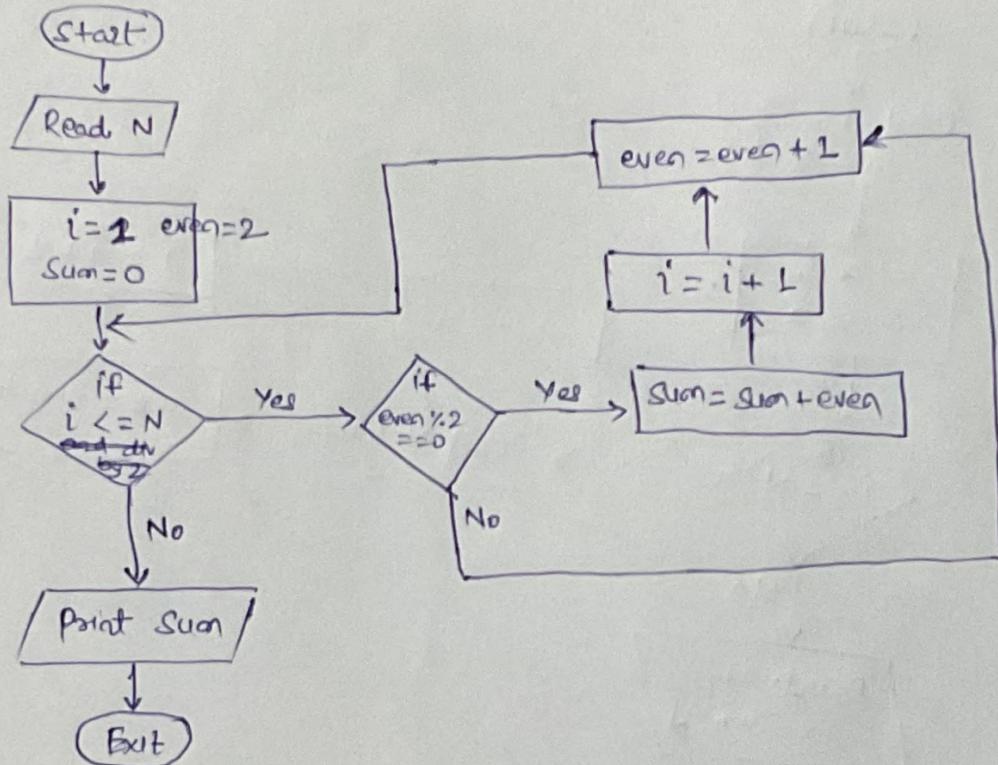
So, the bank security guard is going to wait for the customers outside the bank, if they're coming before 6:00PM, He will add the customer to the queue, otherwise keep on waiting and if time reaches 6PM, He is going to close the bank.

Flowchart Assignment

1. Draw a flowchart to find sum of first N even numbers starting from 2.

Example $N = 4$

$$\text{Output} = 20 \quad (2+4+6+8)$$



Day Run:-

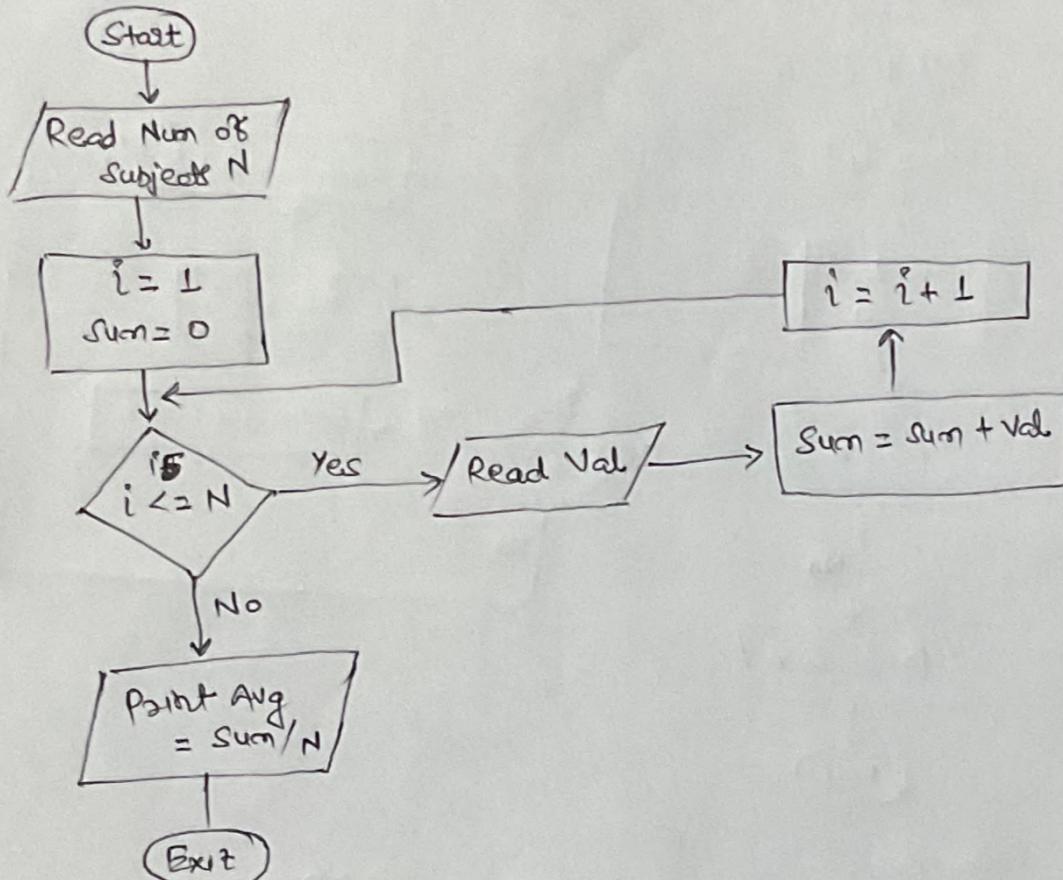
$i = 1$	$N = 4$	$\text{even} = 2$	$\text{sum} = 0$
1	$1 <= 4 \checkmark$	$2 \% 2 == 0 \checkmark$	$0 + 2 = 2$
2	$2 <= 4 \checkmark$	$3 \% 2 == 0 X$	2
2	$2 <= 4 \checkmark$	$4 \% 2 == 0 \checkmark$	$2 + 4 = 6$
3	$3 <= 4 \checkmark$	$5 \% 2 == 0 X$	6
3	$3 <= 4 \checkmark$	$6 \% 2 == 0 \checkmark$	$6 + 6 = 12$
4	$4 <= 4 \checkmark$	$7 \% 2 == 0 X$	12
4	$4 <= 4 \checkmark$	$8 \% 2 == 0 \checkmark$	$12 + 8 = 20$
5	$5 <= 4 X$		
			20

2. Draw a flowchart to input marks in 5 subjects and find their average.

Input: 5

100 80 90 70 90

Output: 86 $\text{Avg} = \frac{100 + 80 + 90 + 70 + 90}{5} = 86$



stry Run

N = 5

		Val	Sum = 0
i = 1	1 ≤ 5 ✓	100	0 + 100 = 100
2	2 ≤ 5 ✓	80	100 + 80 = 180
3	3 ≤ 5 ✓	90	180 + 90 = 270
4	4 ≤ 5 ✓	70	270 + 70 = 340
5	5 ≤ 5 ✓	90	340 + 90 = 430
6	6 ≤ 5 X		

$$\text{Avg} = \frac{430}{5} = \underline{\underline{86}}$$

3. Draw a flowchart to find LCM of two numbers A and B.

You may use the formula $\boxed{LCM \times HCF = \text{Product}(A, B)}$

Input: 6 8

Output: 24

$A = 8$

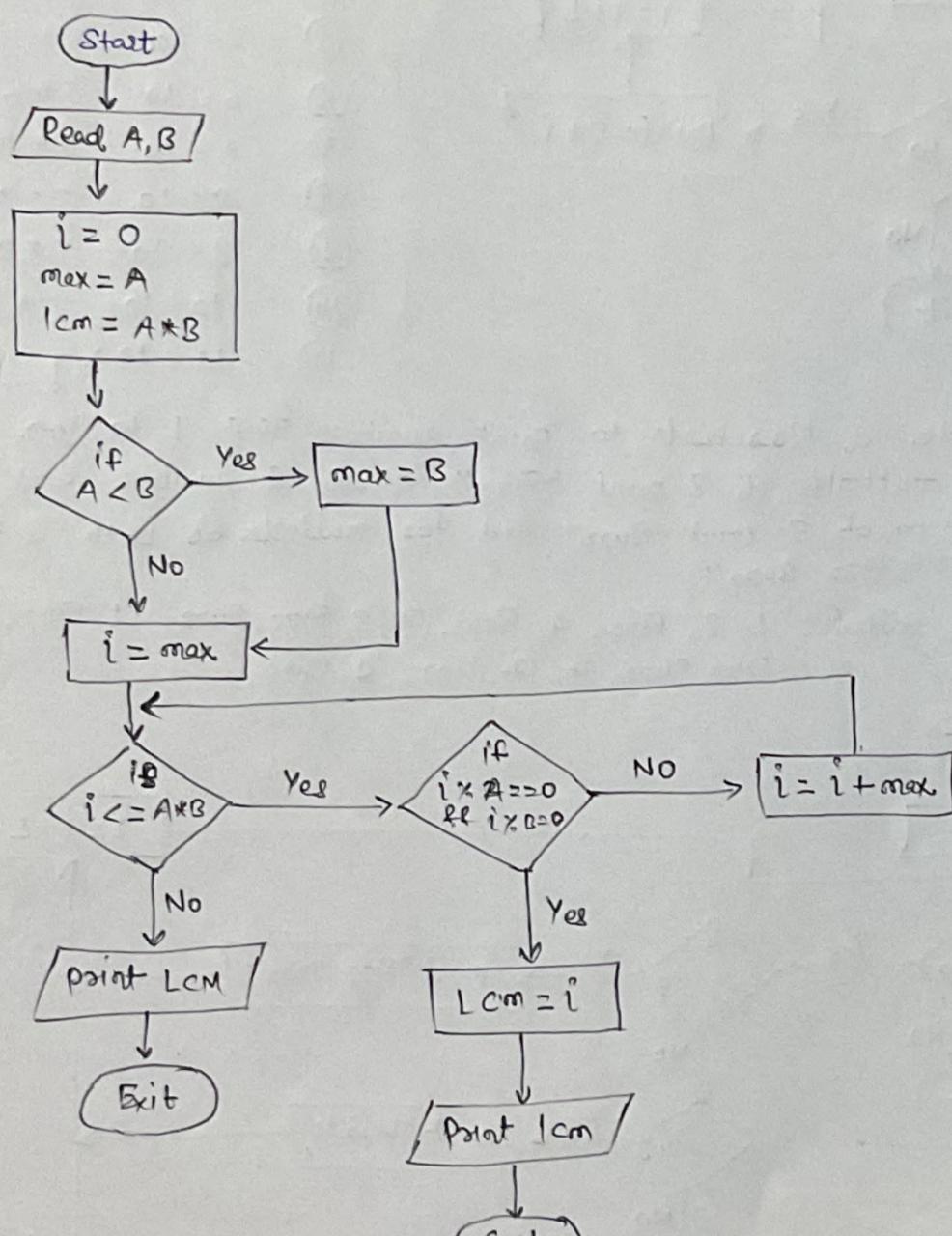
$B = 8$

Lowest Common Multiplication

12	16
18	24
24	32
30	40
36	48
42	
48	

LCM of (A,B) larger betw
max(A,B) to $A \times B$.

8 to 48



Say Run :-

A B

6 8

$i = 0$

$i = 0$

$max = 8$

8

$lcm = 6 * 8 = 48$

$= 48$

$i = 8$

$8 <= 48 \checkmark$

$x \checkmark$

$= 48$

$i = 8 + 8$

$= 16$

$16 \leq 48 \checkmark$

$x \checkmark$

$= 48$

$i = 16 + 8 = 24$

$24 \leq 48 \checkmark$

$v \checkmark$

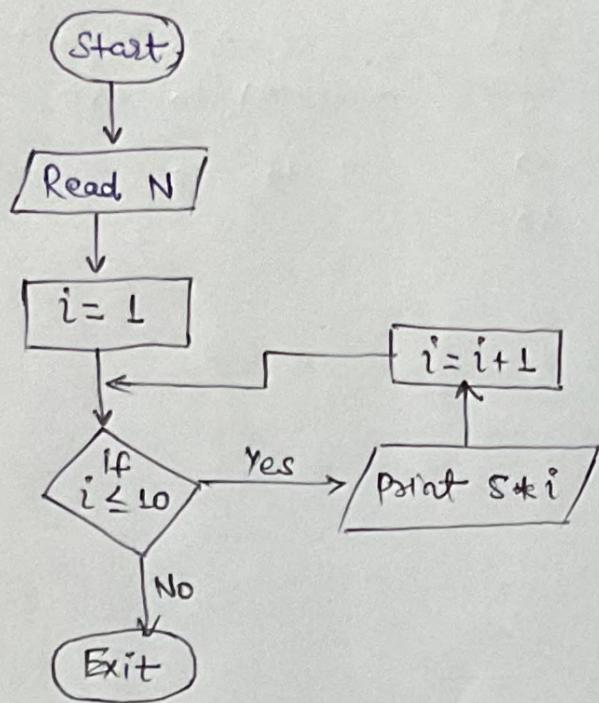
$= 24$

v

4. Draw a flowchart to print table of a given Number upto 10.

Example $N = 5$

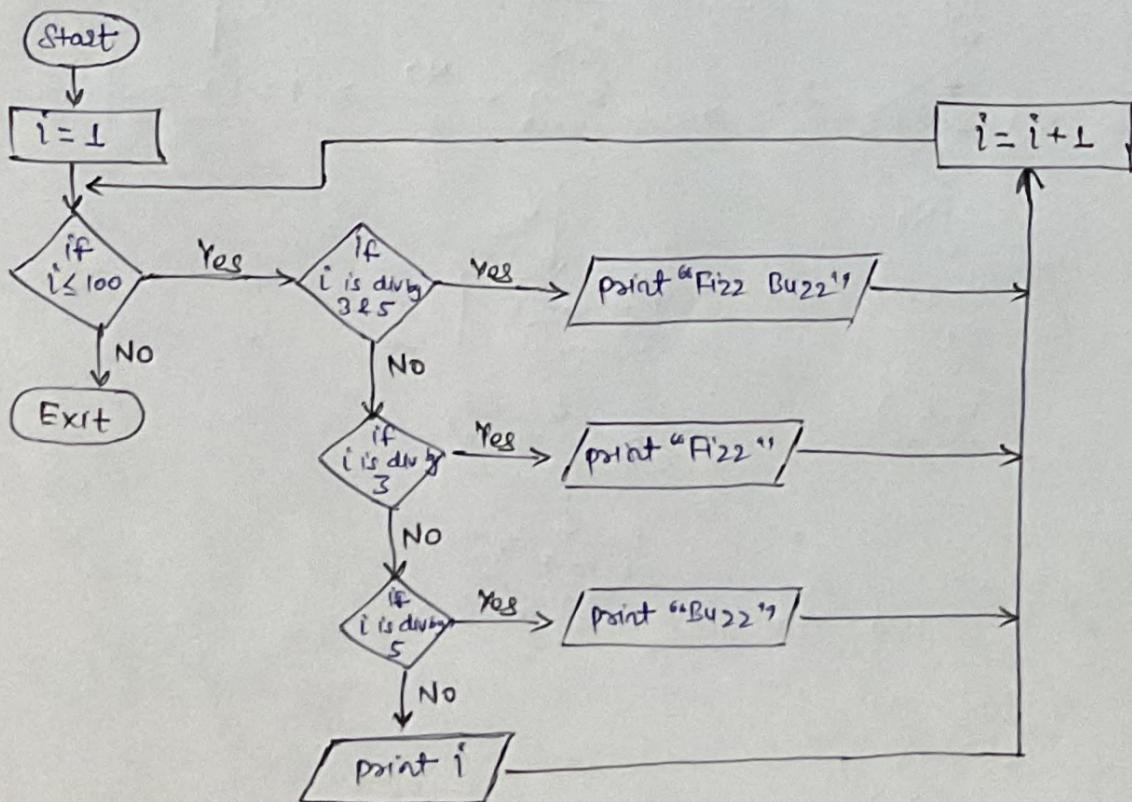
Output : 5, 10, 15, 20, 25, 30, 35, 40, 45, 50,



<u>Dry Run :-</u>		
i = ①	$N = 5$	$1 \leq 10 \vee 5 * 1 = 5$
i = ②		$2 \leq 10 \vee 5 * 2 = 10$
③		$3 \leq 10 \vee 5 * 3 = 15$
④		$4 \leq 10 \vee 5 * 4 = 20$
⑤		$5 \leq 10 \vee 5 * 5 = 25$
⑥		$6 \leq 10 \vee 5 * 6 = 30$
⑦		$7 \leq 10 \vee 5 * 7 = 35$
⑧		$8 \leq 10 \vee 5 * 8 = 40$
⑨		$9 \leq 10 \vee 5 * 9 = 45$
⑩		$10 \leq 10 \vee 5 * 10 = 50$
⑪		$11 \leq 10 \times$

5. Write a flowchart to print numbers from 1 to 100. and for multiples of 3 print "Fizz" instead of number and for multiple of 5 print "Buzz" and for multiple of both 3 & 5 print "Fizz Buzz".

Sample output : 1, 2, Fizz, 4, Buzz, 7, 8, Fizz, Buzz, 11, Fizz, 13, 14, Fizz Buzz, 16, 17, Fizz, 19, Buzz,



Section 3

Logical Thinking - Part 2. Pseudocode

↓
Human readable description
of an algorithm

Pseudocode - Notation.

↓
Sequence of steps to
solve a particular problem

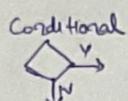
Pseudocode

- You can come up with your own notation of writing pseudocode.
But there are few basic ways of writing it.

Pseudocode → Plain English Text, not a real code that machine can execute.
Why?

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

- Let's define our own 6 types of instructions set



- ✓ • Input [read N]
- ✓ • Assignment [$\&sum=0$] → point "Hello"
- ✓ • Output [point sum] → point sum
- ✓ • If Else [If $i < N$ then end else then end] → if () { } else { }
When we combine multiple boxes it forms loop.
- ✓ • While loop [while $i < N$ do end] →
While $i < N$ {
 loop
 i = i + 1
}
- ✓ • Exit [exit]

① Pseudocode - Simple Interest

Read P, R, T and compute SI

Input

10, 20, 4

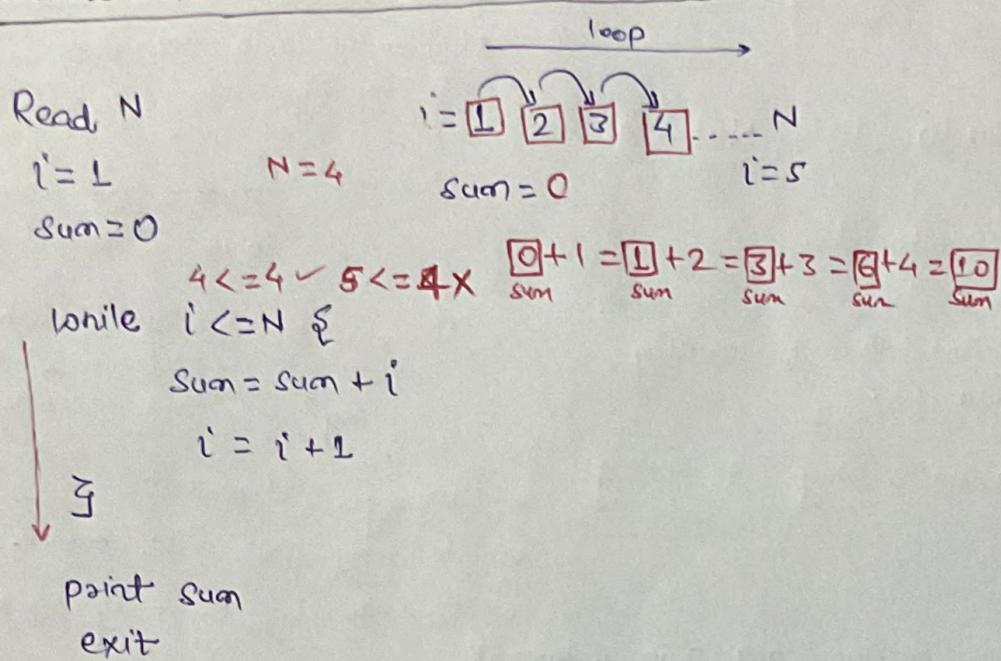
Output

8

- | |
|---------------------------------------|
| 1. read P, R, T |
| 2. $SI = (P \times R \times T) / 100$ |
| 3. point SI |
| 4. exit |

That's it
4 lines of code

① Pseudocode - Sum of Number from L to N



② Pseudocode - Sum of N Numbers

Take input N, followed by N inputs
find the sum of N numbers that have been input.

$N = 4$

10, 20, 40, 30

$sum = 100$

Pseudocode :

Read, N

$i = 1$

$sum = 0$

while $i \leq N \{$

Read, Val

$sum = sum + val$

$i = i + 1$

}

point sum

exit

loop →

10	20	40	30	?
↑	↑	↑	↑	↑
$i=1$	$i=2$	$i=3$	$i=4$	$i=5$

$\boxed{0}$	$\boxed{10}$
sum	val
10	10
30	20
70	40
$\boxed{100}$	30
	sum

Pseudocode - Prime or Not

Pseudocode to check if number is prime or not.

Input

LL

Output

Yes

Read N

i = 2

While $i < N \{$

if $N \% i == 0 \{$

 Print "Not Prime"
 exit

$i = i + 1$

}

Print "Prime"

exit

Dry Run

⑦

i = 2

⑧ $2 < 7$

$7 \% i == 0 \times$

⑨ $3 < 7$

$7 \% 3 == 0 \times$

⑩ $4 < 7$

$7 \% 4 == 0 \times$

⑪ $5 < 7$

$7 \% 5 == 0 \times$

⑫ $6 < 7$

$7 \% 6 == 0 \times$

⑬ $7 < 7 \times$

"Prime"

⑤ $i = 2$ $2 < 9 \vee \times$
 ⑥ $3 < 9 \vee \checkmark$
Not Prime

Pseudocode - GCD

Find GCD of two numbers A and B

A = 8

1

B = 20

GCD = 4

① $2, 3, \dots, \overbrace{\min(A, B)}^{\text{largest}}$

Pseudocode:

Read A, B

i = 1, gcd = 1

While $i \leq \min(A, B) \{$

 if $A \% i == 0 \text{ and } B \% i == 0 \{$

 gcd = i

 }

$i = i + 1$

}

Print gcd

exit

7, 8

	A	B	GCD
i = 1	✓	✓	1
i = 2	✓	✓	2
i = 3	✗	✗	2
i = 4	✓	✓	4
i = 5	✗	✓	4
i = 6	✗	✗	4
i = 7	✗	✗	4
i = 8			

Pseudocode - Star Pattern

$N = 3$

*
* *
* * * *

Observations:

1. There are N rows

2. In i th row there are $\underbrace{\text{num of stars}}_{\text{loop}}$.

Code:

Read N

$i = 1$

while $i \leq N$ {

 Stars = 1

 while Stars $\leq i$ {

 Print "

 Stars = Stars + 1

 Print "\n"

 }

$i = 1 \checkmark$

*

$i = 2 \checkmark$

*, *

$i = 3 \checkmark$

, *, *, *

$i = 4 \times$

}

exit

Pseudocode - Star Pyramid Pattern

Spaces	Stars	$N = 3$	$N-i$
② - - *	1	$3-1=2$	
① - * * *	2	$3-2=1$	
③ * * * *	3	$3-3=0$	

Observation:

$2i-1$

Stars	Row(i)
1	$2(1)-1$
3	$2(2)-1$
5	$2(3)-1$

Observation:

1. There are N Rows

2. There are spaces followed stars in i th row, if you don't want to observe, use predefined formula

Code:

Read N

$i = 1$

Row

while $i \leq N$ {

Spaces = 1

 while Spaces $\leq N-i$ {

 Print " "

 Spaces = Spaces + 1

Stars = 1

 while Stars $\leq 2i-1$ {

 Print "*"

 Stars = Stars + 1

Print "\n" newline

Sequence - [A.P]

1, 3, 5, ..., $\underset{2}{\underset{2}{\overbrace{\dots}}}$ $\underset{i^{\text{th term}}}{\uparrow}$

$$\begin{aligned} T_i &= a + (i-1)d \\ &= 1 + (i-1)2 \\ &= 1 + 2i - 2 \\ &= 2i - 1 \end{aligned}$$

$i = i+1$ update

}

exit

3. Pseudocode to find LCM of two numbers A and B.

You may use formula $LCM * GCD = \text{Product}(A, B)$

Read A, B

$i = 1$

$gcd = 1$

$lcm = A * B$

while $i \leq \min(A, B)$ {

if $A \% i == 0$ and $B \% i == 0$ {

$gcd = i$

}

$i = i + 1$

}

Point $lcm = \frac{A * B}{gcd}$

exit

Read A, B

$i = 0$

$max = A$

$lcm = A * B$

if $A < B$ {

$max = B$

}

$i = max$

while $i \leq A * B$ {

if ($i \% A == 0$ && $i \% B == 0$) {

$lcm = i$

Point lcm

exit

}

$i = i + max$

}

Point lcm

exit.

4. Pseudocode to print table of a given Number upto 10.

$N = 5$

Output : 5, 10, 15, 20, 25, 30, 35, 40, 45, 50

Read N

$i = 1$

while $i \leq 10 \{$

 print $5 * i$

$i = i + 1$

}

exit

5. Pseudocode that prints the numbers from 1 to 100 and for multiple of 3 print "Fizz" instead of number and for multiples of 5 print "Buzz" and for both multiple of 3,5 print "Fizz Buzz".

Sample Output :

1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz, 11, Fizz, 13, 14,
Fizz Buzz, 16, 17, Fizz, 19, Buzz, Fizz, 22, 23,

$i = 1$

while $i \leq 100 \{$

 if $i \% 3 == 0 \& i \% 5 == 0 \{$

 print ("Fizz Buzz")

}

 else if $i \% 3 == 0 \{$

 print ("Fizz")

}

 else if $i \% 5 == 0 \{$

 print ("Buzz")

}

 print i

$i = i + 1$

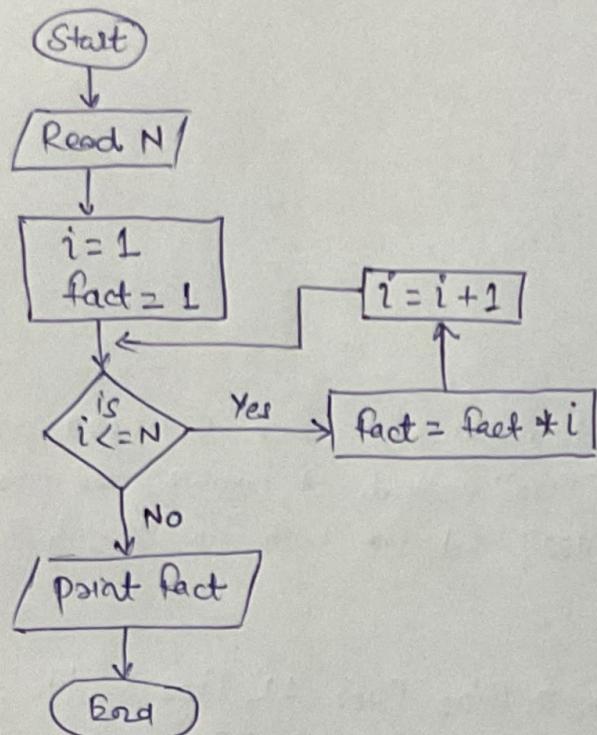
}

exit

6. Pseudocode to find factorial of a number.

$N = 5$

Output : 120 ($1 * 2 * 3 * 4 * 5$)



Try, Run

$N = 5$

i = 1

$$\begin{aligned} \text{fact} &= 1 \\ &= \text{fact} * i \end{aligned}$$

$$\begin{aligned} \text{fact} &= 1 * 1 = 1 \\ &= 1 * 2 = 2 \end{aligned}$$

$$\begin{aligned} \textcircled{1} &<= 5 \checkmark \\ &\text{fact} &= 2 * 3 = 6 \end{aligned}$$

$$\begin{aligned} \textcircled{2} &<= 5 \checkmark \\ &\text{fact} &= 6 * 4 = 24 \end{aligned}$$

$$\begin{aligned} \textcircled{3} &<= 5 \checkmark \\ &\text{fact} &= 24 * 5 = 120 \end{aligned}$$

$$\begin{aligned} \textcircled{4} &<= 5 \checkmark \\ &\text{fact} &= 120 \end{aligned}$$

$$\begin{aligned} \textcircled{5} &<= 5 \checkmark \\ &\text{fact} &= 120 \end{aligned}$$

$$\begin{aligned} \textcircled{6} &<= 5 X \\ &\text{fact} &= 120 \end{aligned}$$

Read N

$i = 1$

$\text{fact} = 1$

while $i <= N \{$

$\text{fact} = \text{fact} * i$

$i = i + 1$

g

print fact

exit

- ⑥ Flowchart - Bank Employee
- ⑦ Flowchart - Bank Security Guard.

Assignment - 1

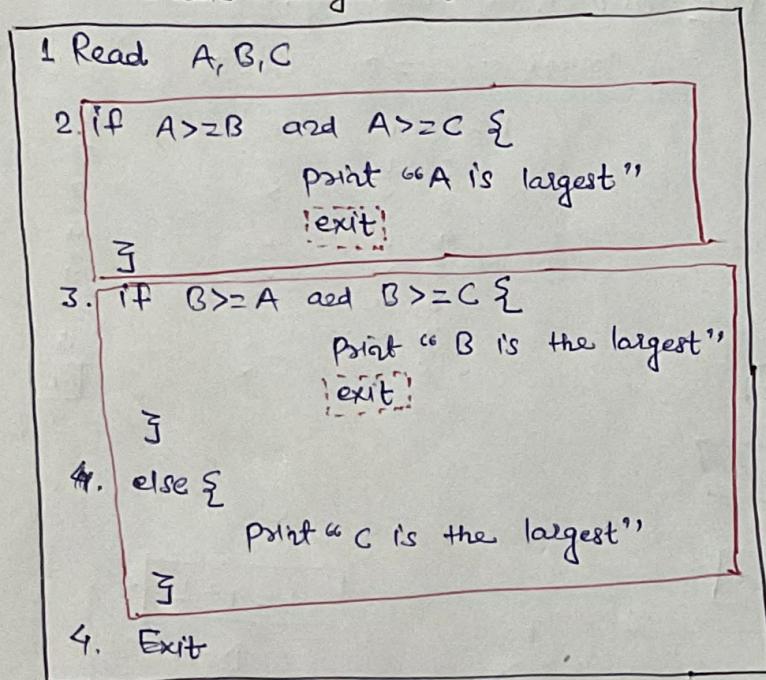
Flowcharts

1. sum of first N even numbers starting from 2.
2. Input 5 subjects and find their Average.
3. LCM of 2 numbers. [$\text{LCM} * \text{GCD} = \text{Product}(A, B)$]
4. Print table of a given Number N.
5. 1 to 100, multiple of 3 = "Fizz" Multiple of 5 = "Buzz" and Multiple of both = "Fizz Buzz". Print the number 1 to 100.
6. Factorial of a Number N.

Pseudocode

Pseudocode - Notation

- ⑧ Pseudocode - Simple Interest.
- ⑨ Pseudocode - Largest of 3 Numbers.



Both blocks are independent to each other that's why exit, or else it will not work properly for the all corner cases.

eg: 20, 10, 5

20 ✓ | 20 ✗
5 ✓ | 20 ✗

if ≡
| ≡
else if ≡
| ≡
else ≡

- ⑩ Sum of number from 1 to N
- ⑪ Pseudocode - Prime or Not.
- ⑫ Pseudocode - Sum of N inputs.
- ⑬ Pseudocode - GCD of 2 numbers.
- ⑭ Pseudocode - Star Pattern . *
- ⑮ Pseudocode - Star Pyramid Pattern

*
* *
* * *
* * * * *

Writing code

⑥ Code Simple Interest

Read P,R,T

$$SI = (P \times R \times T) / 100$$

Print SI

exit

Python is a programming language

Code in Python

```
P = int(input())
R = int(input())
T = int(input())
SI = (P * R * T) / 100
print(SI)
```

si.py

⑦ Largest of 3 Numbers

Read A,B,C

if $A \geq B$ and $A \geq C$ {

 Print A

 exit

if $B \geq A$ and $B \geq C$ {

 Print B

 exit

else

 Print C

 exit

def largestNumber():

A = int(input())

B = int(input())

C = int(input())

if $A \geq B$ and $A \geq C$:

 Print(A)

 return

if $B \geq A$ and $B \geq C$:

 Print(B)

 return

else :

 Print(C)

 return

largest.py

You can convert any Pseudocode into real code.

You just need to know the rule of that programming language which is also known as Syntax of a language.

⑧ Sum of N inputs

Read N

i=1

Sum=0

while $i \leq N$ {

 Read Num

 Sum = Sum + Num

 i = i + 1

}

Print sum

exit

Code

N = int(input())

i = 1

Sum = 0

while $i \leq N$:

 num = int(input())

 Sum = Sum + num

 i = i + 1

Print(Sum)

⑥ Code Prime Number

Read N

$i = 2$

while $i < N \{$

if ($N \% i == 0 \{$

 print NOT PRIME

 exit

$\}$

$i = i + 1$

$\}$

print PRIME

exit

Code

```
def checkPrime():
```

```
    N = int(input())
```

```
    i = 2
```

```
    while i < N:
```

```
        if N % i == 0:
```

```
            print("NOT PRIME")
```

```
            return
```

```
        i = i + 1
```

```
checkPrime()
```

```
    print("PRIME")
```

```
    return
```

```
checkPrime()
```

⑦ Code - Star Pyramid Pattern

Read N

$i = 1$

while $i \leq N \{$

 spaces = 1

 while spaces <= N - i:

 print " "

 spaces = spaces + 1

$\}$

 stars = 1

 while stars <= 2 * i - 1:

 print "*"

 stars = stars + 1

$\}$

 print "\n"

$i = i + 1$

$\}$

exit .

$N = \text{int}(\text{input})$

$i = 1$

while $i \leq N:$

 spaces = 1

 while spaces <= N - i:

 print(" ", end="")

 spaces = spaces + 1

 stars = 1

 while stars <= 2 * i - 1:

 print("*", end="")

 stars = stars + 1

 print()

$i = i + 1$

→ Any pseudocode can be converted to real code very easily.
just you need to know the syntax of language.

① Number Pyramid Pattern

$$N=4$$

	<u>Row</u>	<u>Spaces</u>	<u>Inc Num</u>	<u>Dec Num</u>
- - - 1	i=1	3	1	
- - 2 3 2	i=2	2	2 3	2
- 3 4 5 4 3	i=3	1	3 4 5	4 3
4 5 6 7 6 5 4	i=4	0	4 5 6 7	6 5 4

Observation :

1. There are N rows
2. N-i spaces
3. ith row - i numbers in the inc order
Starting from value i
4. i-1 numbers in the dec order.

