



\*

Software :-

Computer software or just software, is a collection of computer programs and related data that provides the instructions for telling a computer what to do and how to do it.

Any set of instructions that guides the hardware and tells it how to accomplish each task.

Software

System software

Application software

\*

System software :-

System software is computer software designed to operate the computer hardware to provide basic functionality and to provide a platform for running application software.

Refers to the operating system and all utility programs that manage computer resources at a low level.

The BIOS (basic input/output system) gets the computer system started after you turn it on and



Shot on Y12

Vivo AI camera

manages the data flow between the operating system and attached devices such as the hard disk, video adapter, keyboard, mouse and printer.

- The boot programs loads the operating system into the computer's main memory or random access memory (RAM).
- System software also includes system utilities, such as the disk defragmenter and System Restore.

\* Application Software :-

- A program or group of programs designed for end users.
- Allows end user to accomplish one or more specific (non-computer related) tasks.
- \* Examples of Computer Application Software :-
  - Word processor
  - Spreadsheet
  - Presentation Software
  - Database Management System

## Programming Languages

### Computer Program

- A program is a set of instructions following the rules of the chosen language.
- Without programs, computers are useless.
- A program is like a recipe.
- It contains a list of ingredients (called variables) and a list of directions (called statements) that tell the computer what to do with the variables.

### PROGRAMMING LANGUAGE

- Vocabulary and set of grammatical rules (syntax) for instructing a computer to perform specific tasks.
- Programming languages can be used to create computer programs.
- The term programming language usually refers to high-level languages, such as BASIC, C, C++, COBOL, FORTRAN, Ada and Pascal.
- You eventually need to convert your program into machine language so that the computer can understand it.
- There are two ways to do this:
  - Compile the program
  - Interpret the program

\* Compile is to transform a program written in a high-level programming language from source code into object code.

- This can be done by using a tool called compiler.
- A compiler reads the whole source code and translates it into a complete machine code program to perform the required tasks which is output as a new file.

\* Interpreter is a program that executes instructions written in a high-level language.

An interpreter reads the source code one instruction or line at a time, converts this line into machine code and executes it.

## Programming Language

## Computer Programming

→ Computer programming is the process of writing, testing, debugging / troubleshooting and maintaining the source code of computer programs.

This source code is written in a programming language like C++, JAVA, Perl etc.

\* A programmer is someone who writes computer program.

Computers programmers write, test and maintain

programs or software that tell the computer what to do.

B. What skills are required to become a programmer?

- Programming :- Writing computer programs for various purposes.
- Writing :- Communicating effectively with others in writing as indicated by the needs of the audience.
- Reading Comprehension :- Understanding written sentences and paragraphs in work-related documents.
- Critical Thinking :- Using logic and analysis to identify the strengths and weaknesses of different approaches.
- Computer and Electronics :- Knowledge of electronic circuit boards, processors, chips and computer hardware and software, including applications and programming.
- Mathematics :- Knowledge of numbers, their operations, and interrelationships.

including arithmetic, algebra, geometry, calculus, statistics, and their applications.

- Oral Expression:- The ability to communicate information and ideas in speaking so others will understand.
- Oral Comprehension:- The ability to listen to and understand information and ideas presented through spoken words and sentences.
- Written Expression:- The ability to communicate information and ideas in writing so others will understand.
- Writing Comprehension:- The ability to read and understand information and ideas presented in writing.
- Deductive Reasoning:- The ability to apply general rules to specific problems to come up with logical answers. It involves deciding if an answer makes sense.
- Information Organization:- Finding ways to structure and classify multiple pieces of information.

\*

## Generations of Programming Language

- The first generation languages, or 1GL, are low-level languages that are machine language.
- The second generation languages, or 2GL, are also low-level languages that generally consist of assembly languages.
- The third generation languages, or 3GL, are high-level languages such as C.
- The forth generation languages, or 4GL, are languages that consist of statements similar to statements in a human language. Fourth generation languages are commonly used in database programming and scripts.
- The fifth generation languages, or 5GL, are programming languages that contain visual tools to help develop a program. A good example of a fifth generation language is Visual Basic.

## \* Types of Programming Language

→ There are three types of programming language:

- (i) Machine language (Low-level language)
- (ii) Assembly language (Low-level language)
- (iii) High-level language

Low-level languages are closer to the language used by a computer, while high-level languages are closer to human languages.

## \* Machine Language

- Machine language is a collection of binary digits or bits that the computer reads and interprets.
- Machine languages are the only languages understood by computers.
- While easily understood by computers, machine languages are almost impossible for humans to use because they consist entirely of numbers.

### Machine language

169 1 160 0 153 0 128 153 0 129 153 130 153  
0 131 900 208 941 96

### High Level language

FOR I = 1 TO 1000: PRINT "A";: NEXT I

### \* Example :-

- Let us say that an electric toothbrush has a processor and main memory.
- The processor can rotate the bristles left and right and can check the on/off switch.
- The machine instructions are one byte long, and correspond to the following machine operations:

Machine Instruction	Machine Operation
0000 0000	Stop
0000 0001	Rotate bristles left
0000 0010	Rotate bristles right
0000 0100	Go back to start of program
0000 1000	Skip next instruction if switch is off
<del>0000</del>	

### \* Assembly Language

- A program written in assembly language consists of series of instructions mnemonics that correspond to a stream of executable instructions, when translated by an assembler, that can be loaded into memory and executed.
- Assembly languages use keywords and symbols, much like English, to form a programming language but at the same time introduce a new problem.

- The problem is that the computer doesn't understand the assembly code, so we need a way to convert it to machine code, which the computer does understand.
- Assembly language programs are translated into machine language by a program called an assembler.

#### \* Example :

- Machine language :

10110000 01100001

- Assembly language :

mov al, #061h

- Meaning :

Move the hexadecimal value 61 (97 decimal) into the processor register named "al".

#### \* High-Level Language

- High-level languages allow us to write computer code using instructions resembling everyday spoken language (for example: print, if, while) which are then translated into machine language to be executed.

Programs written in a high-level language need to be translated into machine language before they can be executed.

Some programming languages use a compiler to perform this translation and others use an interpreter.



*	Examples :- ADA C C++ JAVA BASIC COBOL PASCAL PHYTOM
---	---

### \* Comparison

Machine language	Assembly language	High-level language
Time to execute  Since it is the basic language of the computer, it does not require any translation and hence ensures better machine efficiency. This means the programs run faster.	A program called an 'assembler' is required to convert the program into machine language. Thus, it takes longer to execute than a machine language program.	A program called a compiler or interpreter is required to convert the program into machine language. Thus, it takes more time for a computer to execute.
Time to develop  Needs a lot of skill, as instructions are very lengthy and complex. Thus, it takes more time to program.	Simpler to use than machine language, though instruction codes must be memorized. It takes less time to develop programs as compared to machine language.	Easiest to use. Takes less time to develop programs and, hence, ensures better program efficiency.

\* Algorithm :- Algorithm is a step-by-step procedure which is helpful in solving a problem. If, it is written in English like sentences then, it is called as 'PSEUDO CODE'.

### \* Properties of an Algorithm

An algorithm must possess the following five properties:

- Input
- Output
- Finiteness
- Definiteness
- Effectiveness

### \* Example :-

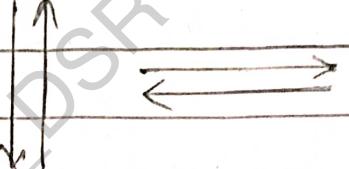
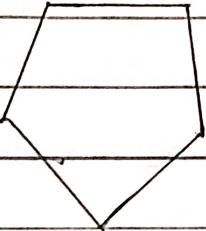
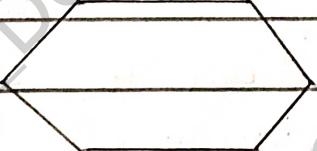
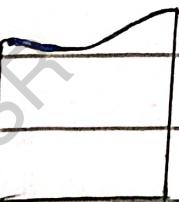
Algorithm for finding the average of three numbers is as follows—

- Start
- Read 3 numbers  $a, b, c$
- Compute sum =  $a + b + c$
- Compute average =  $\text{sum}/3$
- Print average value
- Stop

\* Flowchart :- Diagrammatic representation of an algorithm is called flow chart.

Symbols used in flowchart are mentioned below:

Name	Symbol	Purpose
Terminal		Start/stop / begin/end
Oval		
Input/Output		Input/Output of data
Parallelogram		
Process		Any processing to be performed can be represented.
Rectangle		
Decision box		Decision operation that determine which of the alternative paths to be followed
Diamond		
Connector		Used to connect different parts of flowchart
Circle		

Flow		Join 2 symbols and also represents flow of execution.
Pre defined process		Module (or) subroutines specified else where.
	Double sided rectangle	
Page connection		Used to connect flowchart in 2 different pages.
	Pentagon	
For loop symbol		Shows initialization, condition and incrementation of loop variable.
	Hexagon	
Document		Shows the data that is ready for printout
	Printout	

## \* Difference b/w Algorithm and flowchart

Parameters	Flowchart	Algorithm
Description	A flowchart is a graphical representation of the steps a program takes to process data. In this we can use several geometric patterns to illustrate the numerous actions the program carries out.	An algorithm is a procedure or set of rules that defines how a program is to be executed. Or, we can say that it is a set of instructions for solving a well-defined computational problem.
complexity	It is easy to design and also very user friendly.	It is comparatively difficult to create and also a bit challenging to be understood by a layman.
Geometrical diagrams	It utilizes different types of geometrical shapes, symbols and patterns.	An algorithm does not include any sort of geometrical pattern.
Scope of usage	It can be used in different disciplines to describe a process.	It is used in the domain of mathematics and computer science.



Use	It is used in documenting, designing and analyzing a program in different disciplines.	It is used to represent the concept of decidability.
Users	It doesn't demand the knowledge of a computer programming language.	It demands the knowledge of a computer programming language.
Debugging	It is easy to debug the errors in flowcharts.	It is difficult to debug the errors in algorithms.
Implementation	In flowcharts, no rules are used.	In algorithms, predefined rules are used.
Branching and Looping	Simple to display branching and looping.	Hard to display branching and looping.
Solution	In a flowchart, the solution is represented in a graphical format.	In an algorithm, the solution is presented in non-computer language.



Shot on Y12

Vivo AI camera

2022.04.10 11:04

### Algorithm :-

Q.1 Write an algorithm to find the addition of two numbers.

#### Method - I

Step 1: Start

Step 2: Read a, b

Step 3:  $c = a + b$

Step 4: Print c

Step 5: Stop

#### Method - II

Step 1: Start

Step 2: Take two numbers

Step 3: Add both number and store the result.

Step 4: Show result

Step 5: Stop

Q.2 Write an algorithm to find the addition and multiplication of three numbers.

Step 1: Start

Step 2: Read a, b, c

Step 3: sum =  $a + b + c$

Step 4: Mul =  $a * b * c$

Step 5: Print sum, mul

Step 6: Stop

Q.8

Write an algorithm to find simple interest.

Step 1: Start

Step 2: Read  $p, r, t$

Step 3:  $S.I. = p \times r \times t / 100$

Step 4: Print  $S.I.$

Step 5: Stop

(Ans)

① Write an algorithm to find compound interest.

Step 1: Start

Step 2: Read  $p, r, n$

Step 3: Calculate  $C.I. = p \times (1 + r/100)^n - p$

Step 4: Print  $C.I.$

Step 5: Stop.

② Write an algorithm to find the area of triangle using heron's formula.

Step 1: Start

Step 2: Read  $a, b, c$

Step 3:  $s = (a+b+c)/2$

Step 4:  $A = \sqrt{s \times (s-a) \times (s-b) \times (s-c)}$

Step 5: Print  $A$

Step 6: Stop



Shot on Y12

Vivo AI camera

2022.04.10 11:04

(3)

Write an algorithm to find the area of a circle.

Step 1 : Start

Step 2 : Read  $\pi$

Step 3 :  $A = \pi \times r \times r$

Step 4 : Print A'

Step 5 : Stop

(4)

Write an algorithm to find the addition, multiplication, average of 4 numbers.

Step 1 : Start

Step 2 : Read a, b, c, d

Step 3 : sum = a + b + c + d

Step 4 : Mul = a \* b \* c \* d

Step 5 : Average = sum / 4

Step 6 : Print sum

Step 7 : Print mul

Step 8 : Print average

Step 9 : Stop

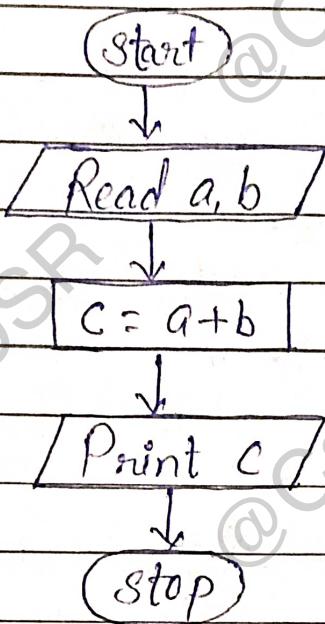


Shot on Y12  
Vivo AI camera

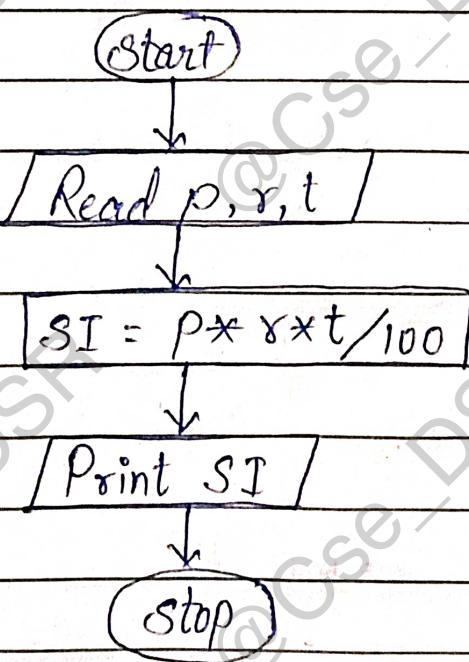
2022.04.10 11:04

Flowchart:-

- Q.1. Draw the flowchart to find the addition of two numbers.



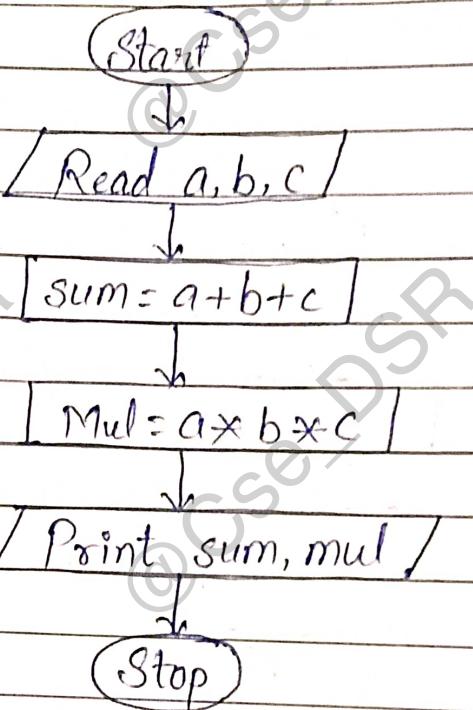
- Q.2. Draw the flowchart to find the simple interest.



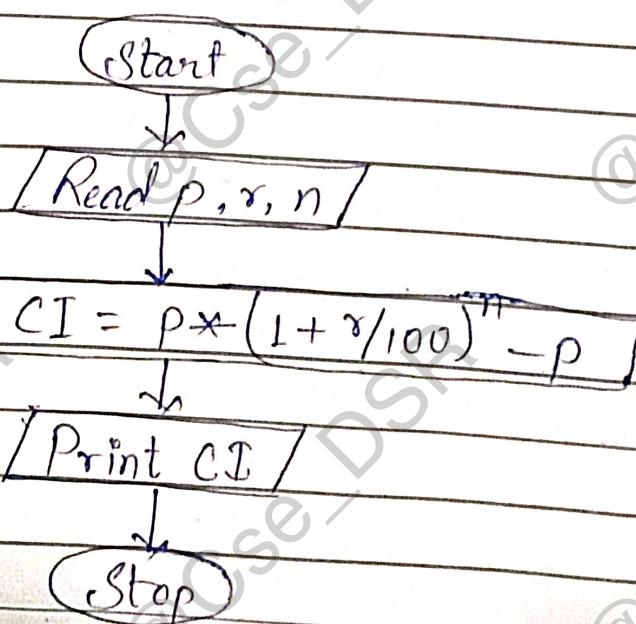
Shot on Y12

Vivo AI camera

Q.3. Draw the flowchart to find the addition and multiplication of three numbers.



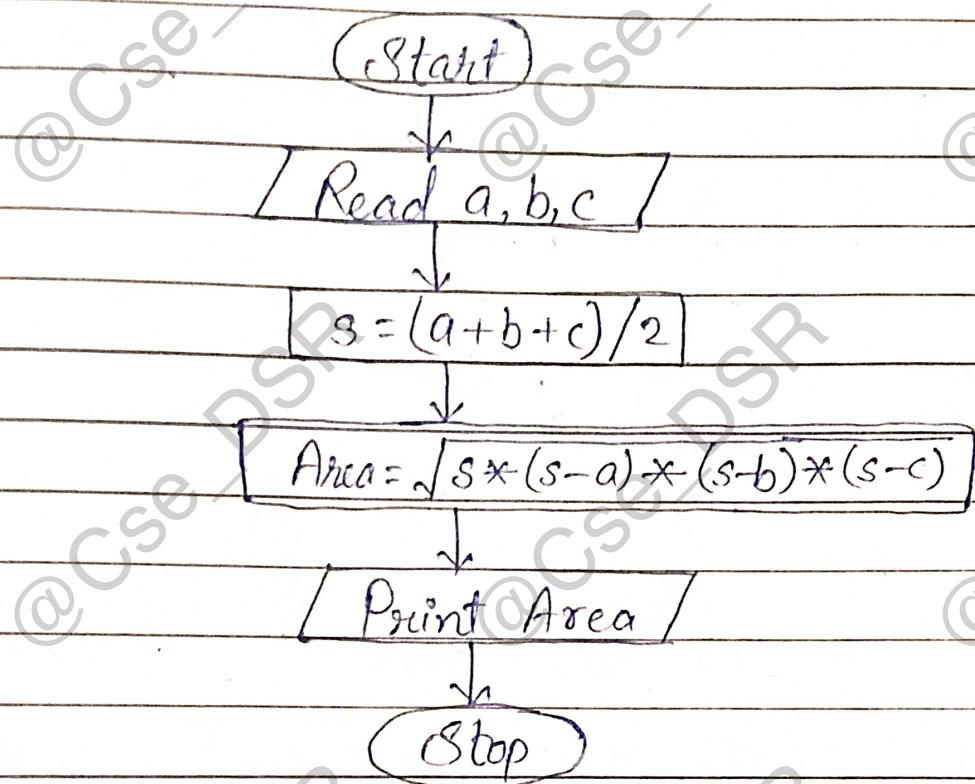
Q.4. Draw the Flowchart to find compound interest.



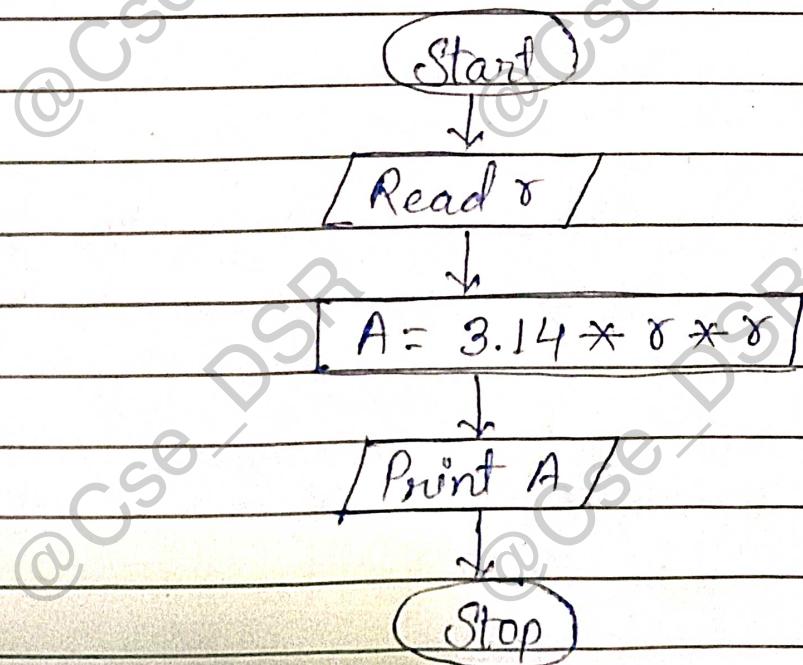
Shot on Y12  
Vivo AI camera

2022.04.10 11:04

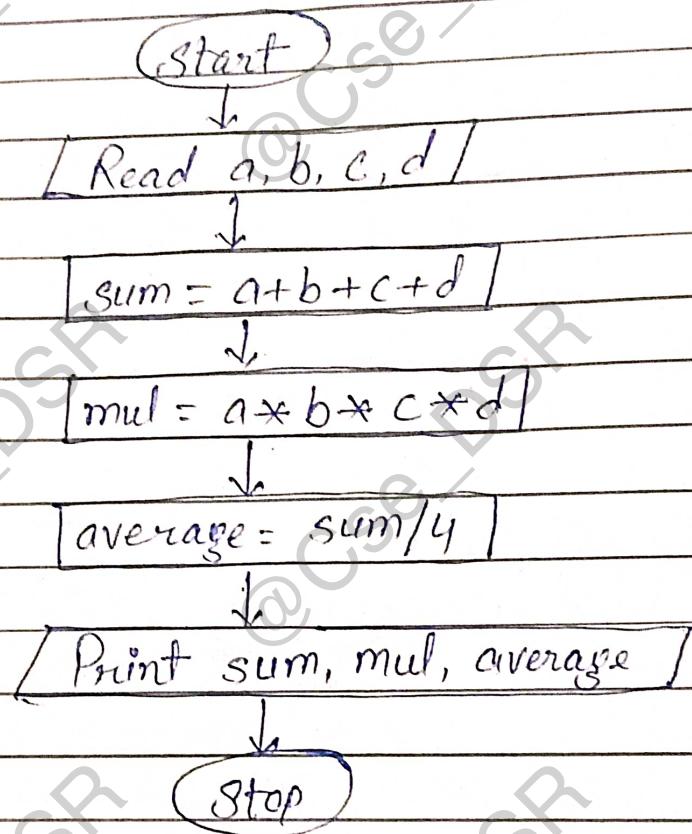
Q.6 Draw the flowchart to find the area of triangle using heron's formula.



Q.6 Draw the flowchart to find the area of circle.



Q.7 Draw the flowchart to find the addition, multiplication and average of 4 numbers.



### \* Conditional Algorithm :-

Q.1. Write an algorithm and draw the flowchart to find biggest between two numbers.

Step 1 : Start

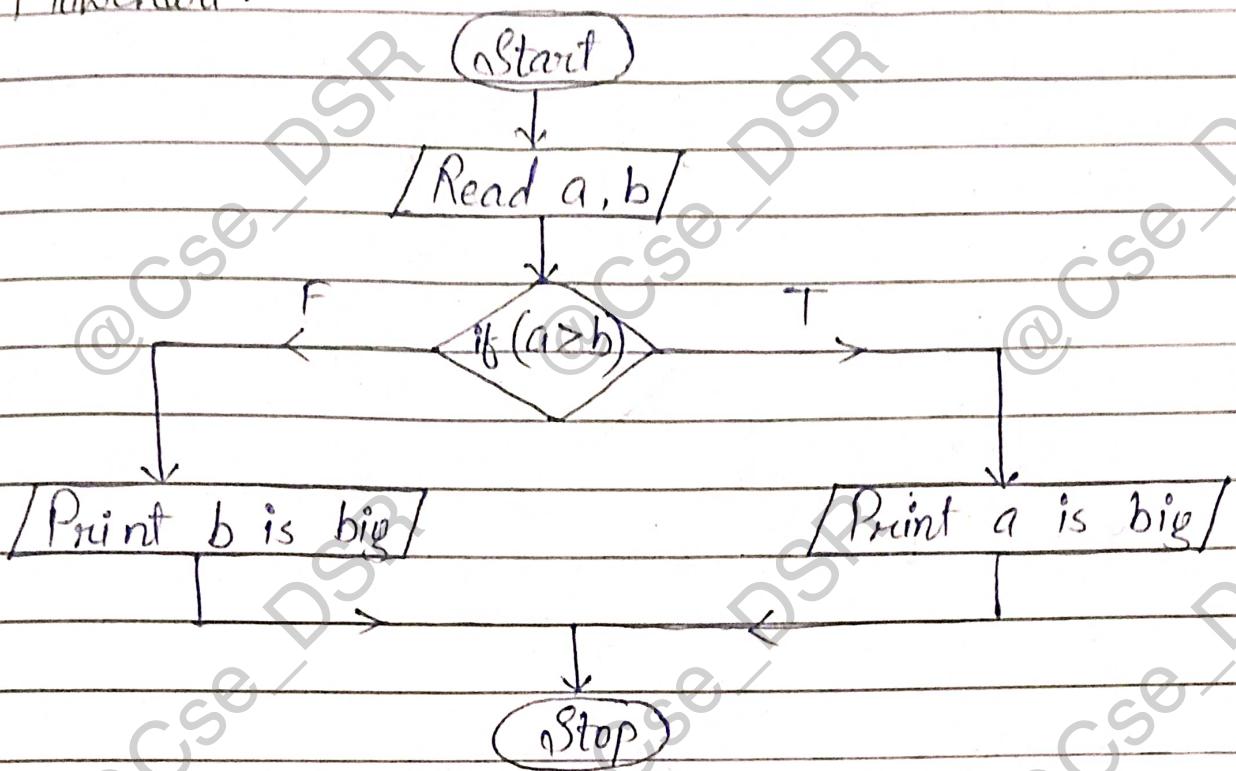
Step 2 : Read a, b

Step 3 : if ( $a > b$ )

print a is big  
else

print b is big  
Step 4 : Stop

Flowchart:-



Q2. Write an algorithm and draw the flowchart to find smallest between two numbers.

Step 1: Start

Step 2: Read a, b

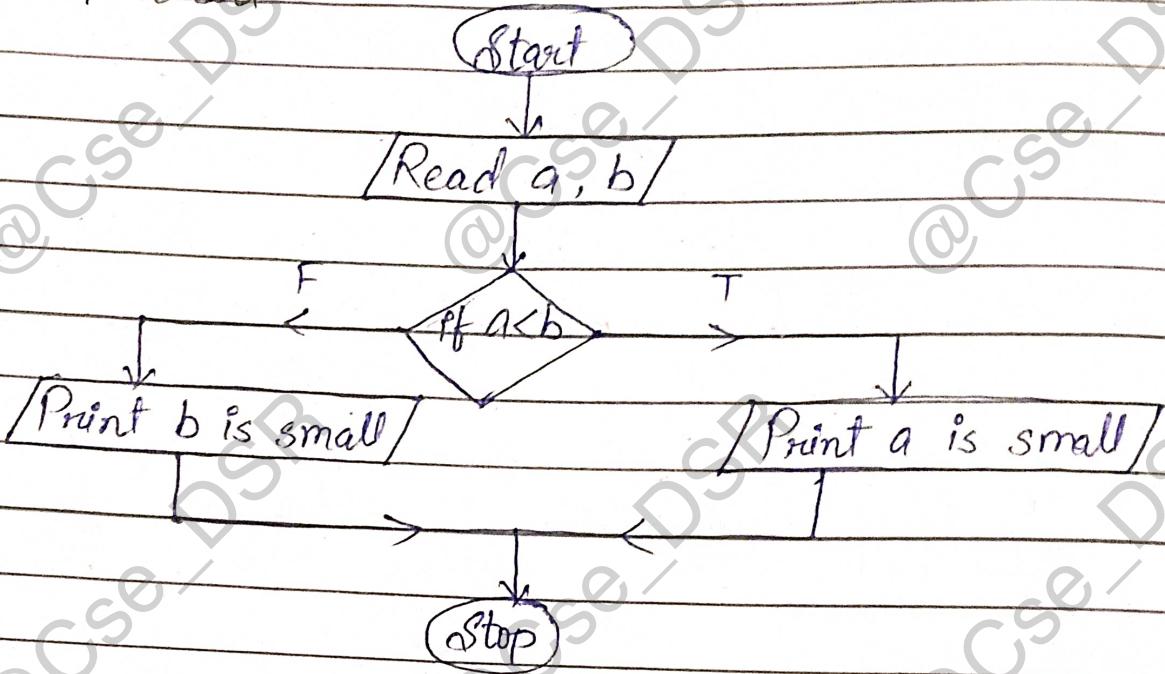
Step 3: if ( $a < b$ )  
print a is small

else

print b is small

Step 4: Stop

Flowchart:-



Q.3. Write an algorithm and draw the flowchart to find smallest between three numbers.

Step 1: Start

Step 2: Read a, b, c

Step 3: if (a < b)

{  
  if (a < c)

    print a is small

  else

    print c is small

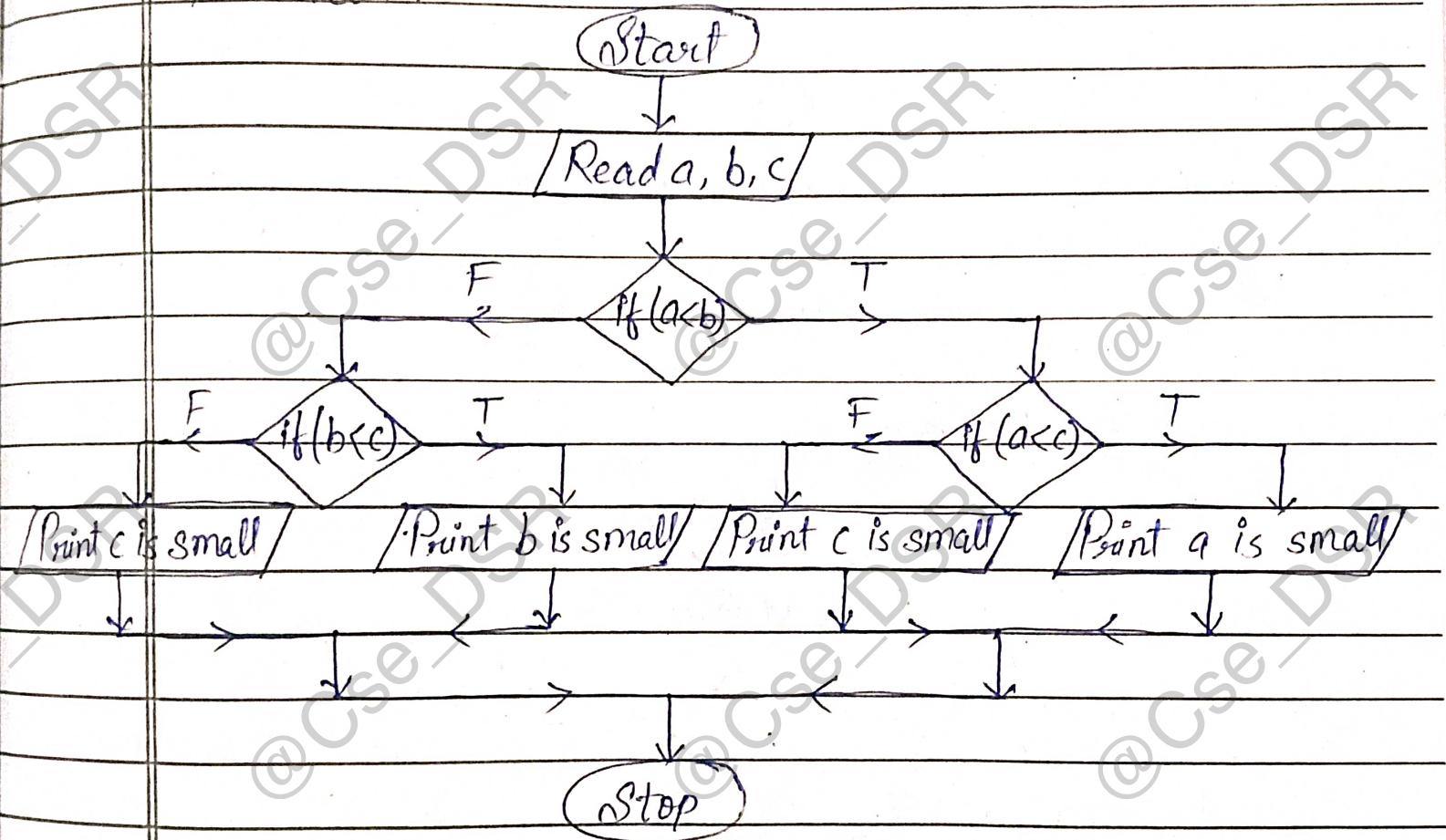
  else

    {  
      if (b < c)

        if (b < c)

print b is small  
 else  
 } print c is small  
 Step 4: Stop

Flowchart:-



Q4. Write an algorithm and draw the flowchart to find greatest between three numbers.

Step 1: Start

Step 2: Read a, b, c

Step 3: if ( $a > b$ )

if ( $a > c$ )  
print a is big

else  
print c is big

}

else

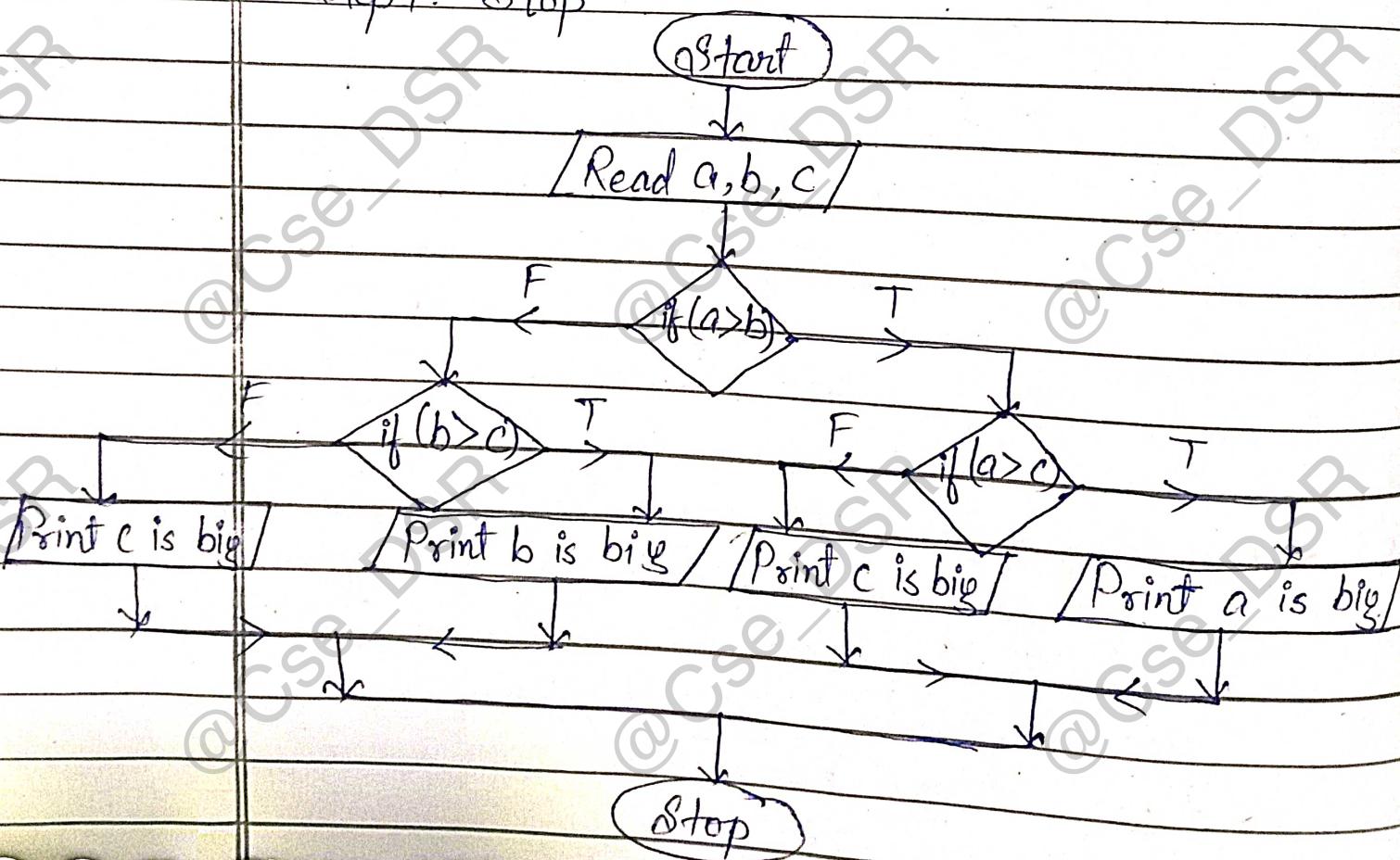
if ( $b > c$ )  
print b is big

else

print c is big

}

Step 4: Stop



## \* Looping Algorithm :-

Q.1. Write an algorithm and draw the flowchart to print your name n times.

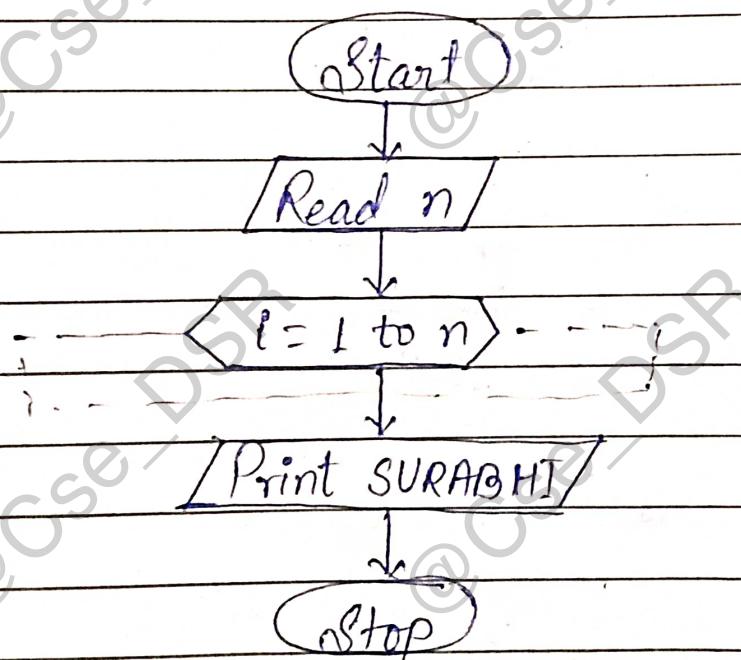
Step 1: Start

Step 2: Read n

Step 3: Loop  $i = 1$  to  $n$

            print SURABHI

Step 4: Stop



Q.2. Write an algorithm and draw the flowchart to find the sum of given series :-

$1 + 2 + 3 + 4 + \dots$  upto  $n$  terms.

Step 1: Start

Step 2: Read n

Step 3: sum = 0

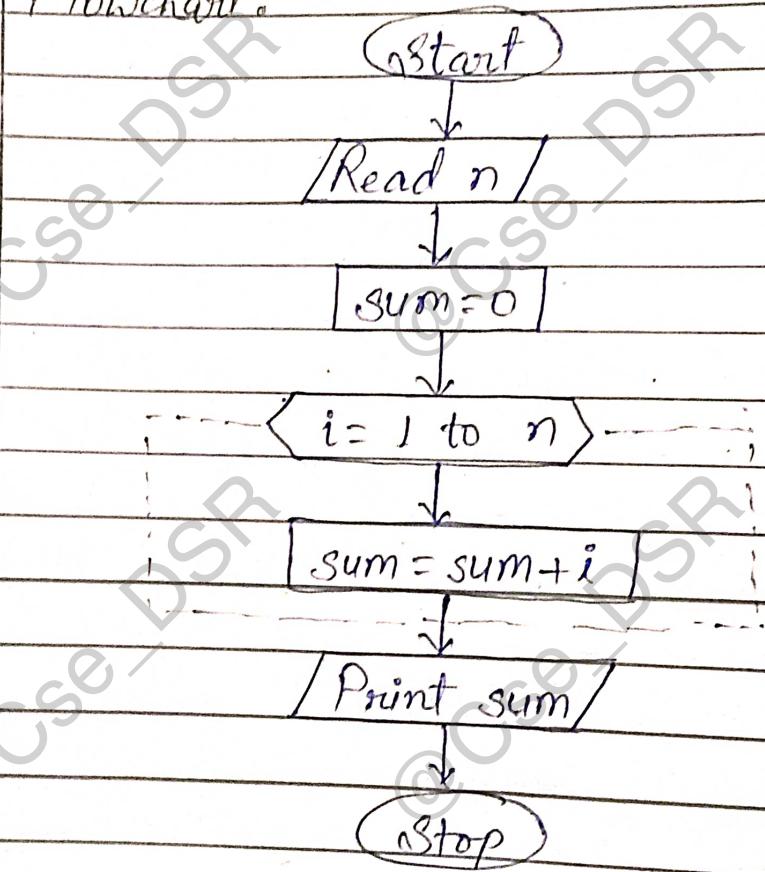
Step 4: loop i = 1 to n

$$\text{sum} = \text{sum} + i$$

Step 5: Print sum

Step 6: Stop

Flowchart:-



Q.3.

Write an algorithm and draw the flowchart to find the factorial of any given number.

Step 1: Start

Step 2: Read n

Step 3: mul = 1

Shot on Y12  
Vivo AI camera

2022.04.10 11:05

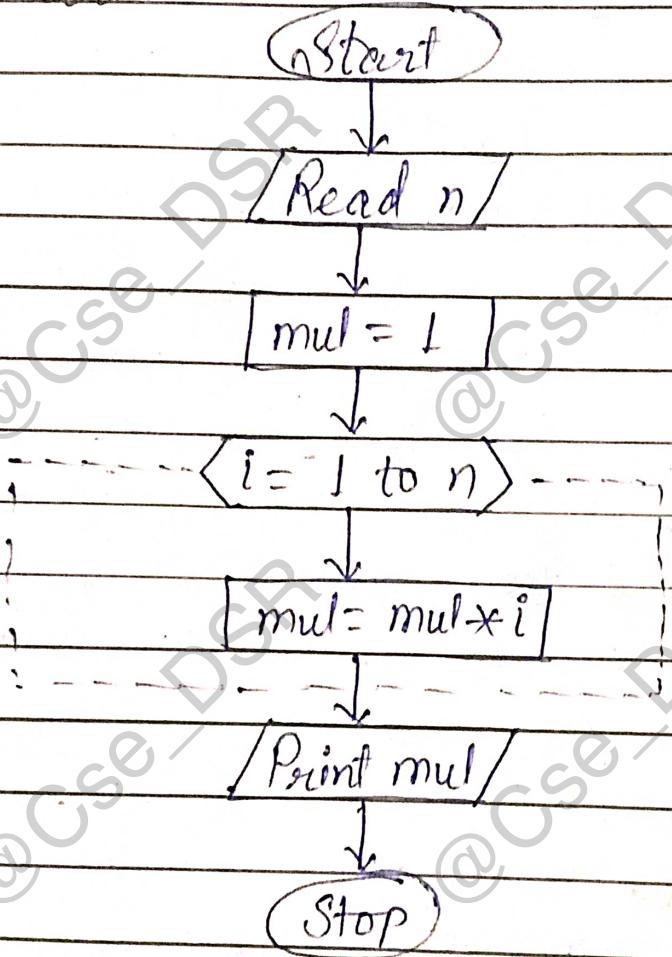
Step 4 : Loop  $i = 1$  to  $n$

$mul = mul \times i$

Step 5 : Print mul

Step 6 : Stop

Flowchart:-



Dry run :-

$$n = 4$$

$$mul = 1$$

$$i = 1 \text{ to } 4$$

$$i = 1$$

$$1 \leq 4$$

$$mul = mul \times i$$

$$i = 2 \quad 2 \leq 4$$

$$\text{mul} = \text{mul} \times i \\ = 1 \times 2 = 2$$

$$i = 3 \quad 3 \leq 4$$

$$\text{mul} = \text{mul} \times i \\ = 2 \times 3 = 6$$

$$i = 4 \quad 4 \leq 4$$

$$\text{mul} = \text{mul} \times i \\ = 6 \times 4 = 24$$

$$i = 5 \quad 5 \leq 4 \times$$

24

Q4. Write an algorithm and draw the flowchart to check given number is prime or not.

Step 1: Start

Step 2: Read n

Step 3: count = 0

Step 4: loop  $i = 1$  to  $n$

if ( $n \% i == 0$ )

count = count + 1

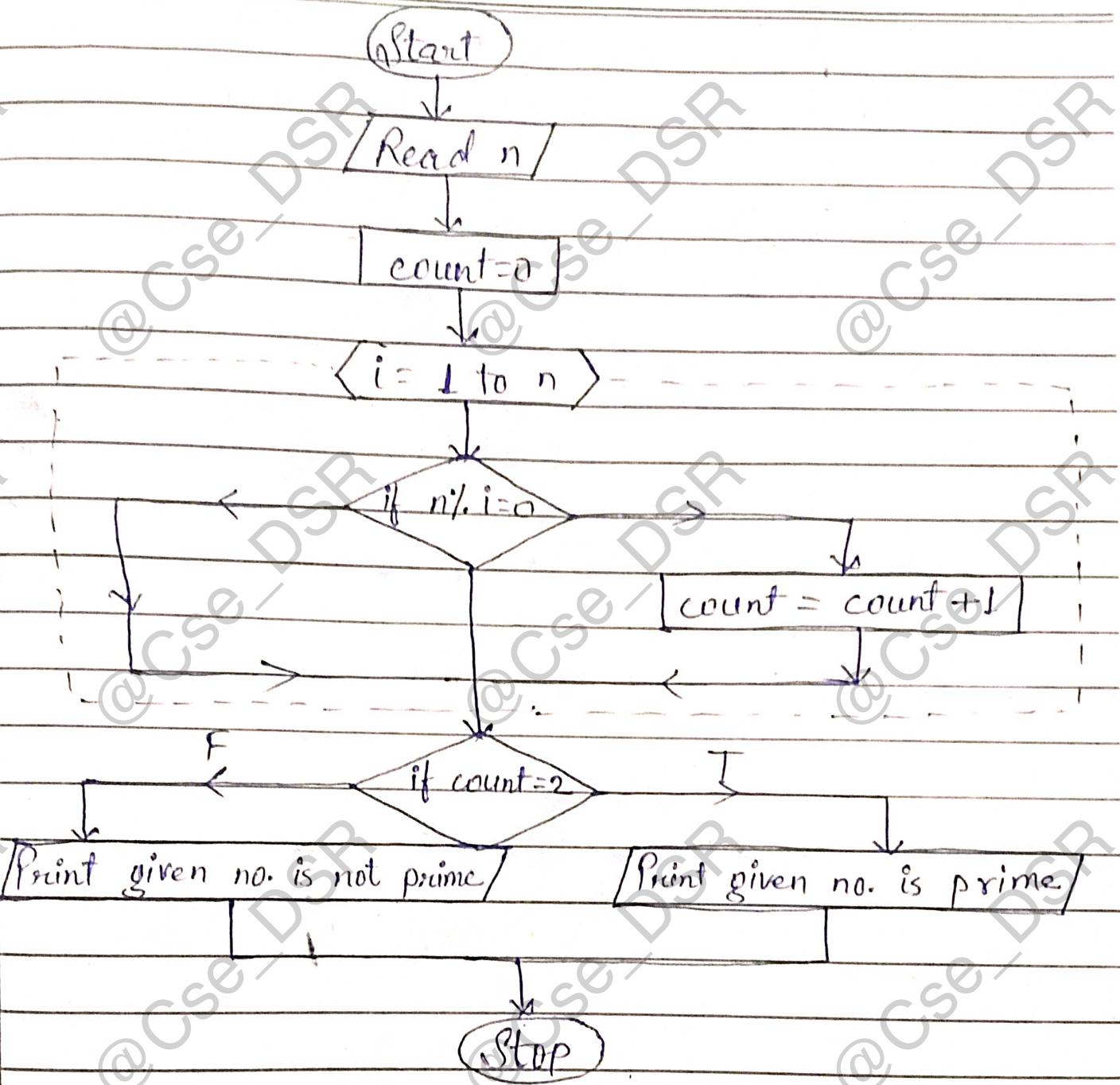
Step 5: if (count = 2)

print given number is prime.

else

print given number is not prime.

Step 6: Stop



Q5. Write an algorithm and draw the flowchart to print reverse number of any given number.

Step 1: Start

Step 2: Read  $n$

Step 3:  $rev = 0$

Step 4: loop ( $n > 0$ )

$$d = n \% 10$$

$$\text{rev} = \text{rev} \times 10 + d$$

$$n = n / 10$$

Step 5: Print rev

Step 6: Stop

Dry run:-

$$n = 123$$

$$123 > 0$$

$$d = 123 \% 10 = 3$$

$$\text{rev} = 0 \times 10 + 3 = 3$$

$$n = 123 / 10 = 12$$

$$12 > 0$$

$$d = 12 \% 10 = 2$$

$$\text{rev} = 3 \times 10 + 2 = 32$$

$$n = 12 / 10 = 1$$

$$1 > 0$$

$$d = 1 \% 10 = 1$$

$$\text{rev} = 32 \times 10 + 1 = 321$$

$$n = 1 / 10 = 0$$

$$0 > 0 \quad \times$$

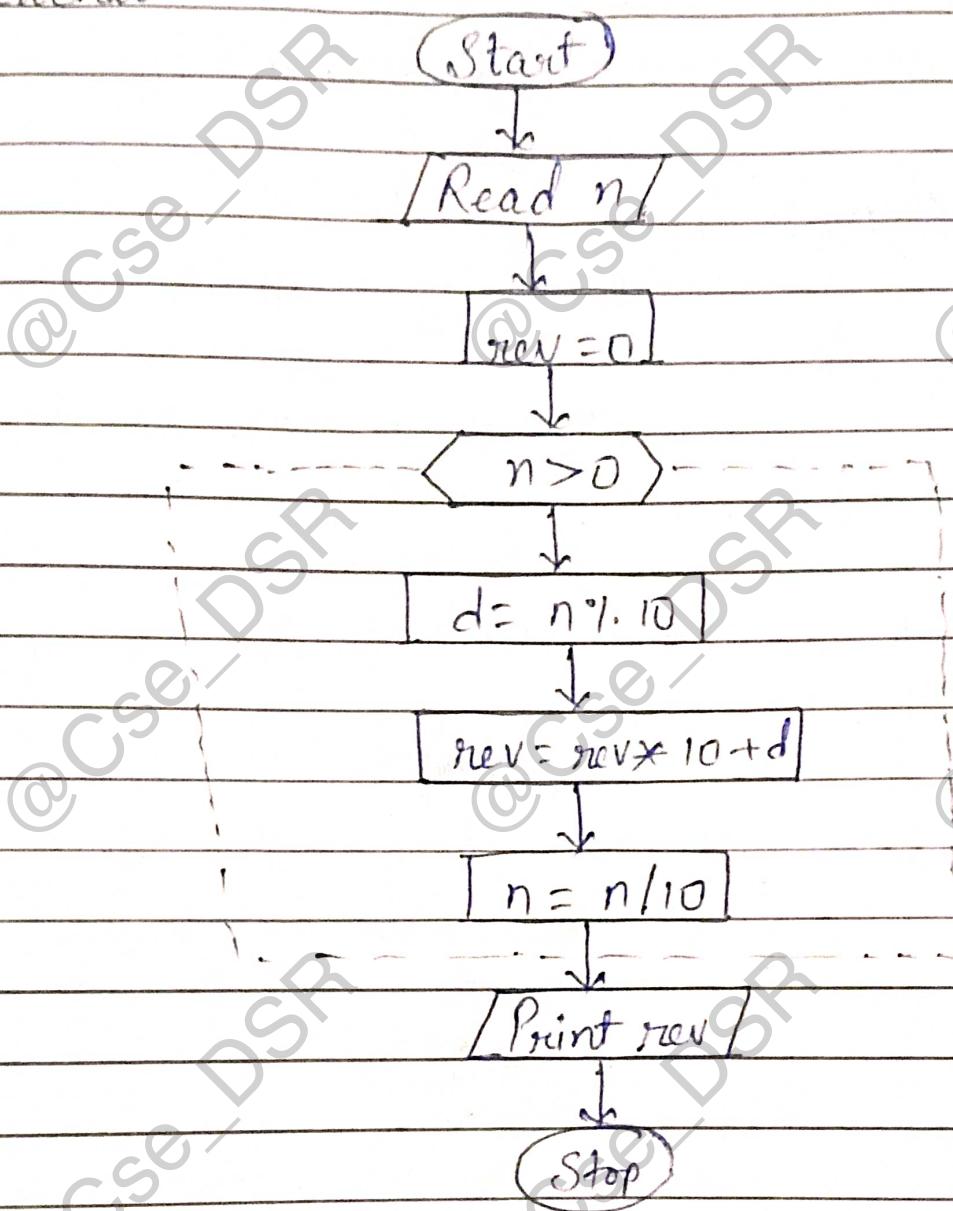
$$321 \quad \cancel{x}$$



Shot on Y12  
Vivo AI camera

2022.04.10 11:05

Flowchart:-



Q.6. Write an algorithm and draw the flowchart to check given number is palindrome or not:

Step 1: Start

Step 2: Read n

Step 3: rev = 0

Step 4: m = n

Step 5: Loop ( $n > 0$ )

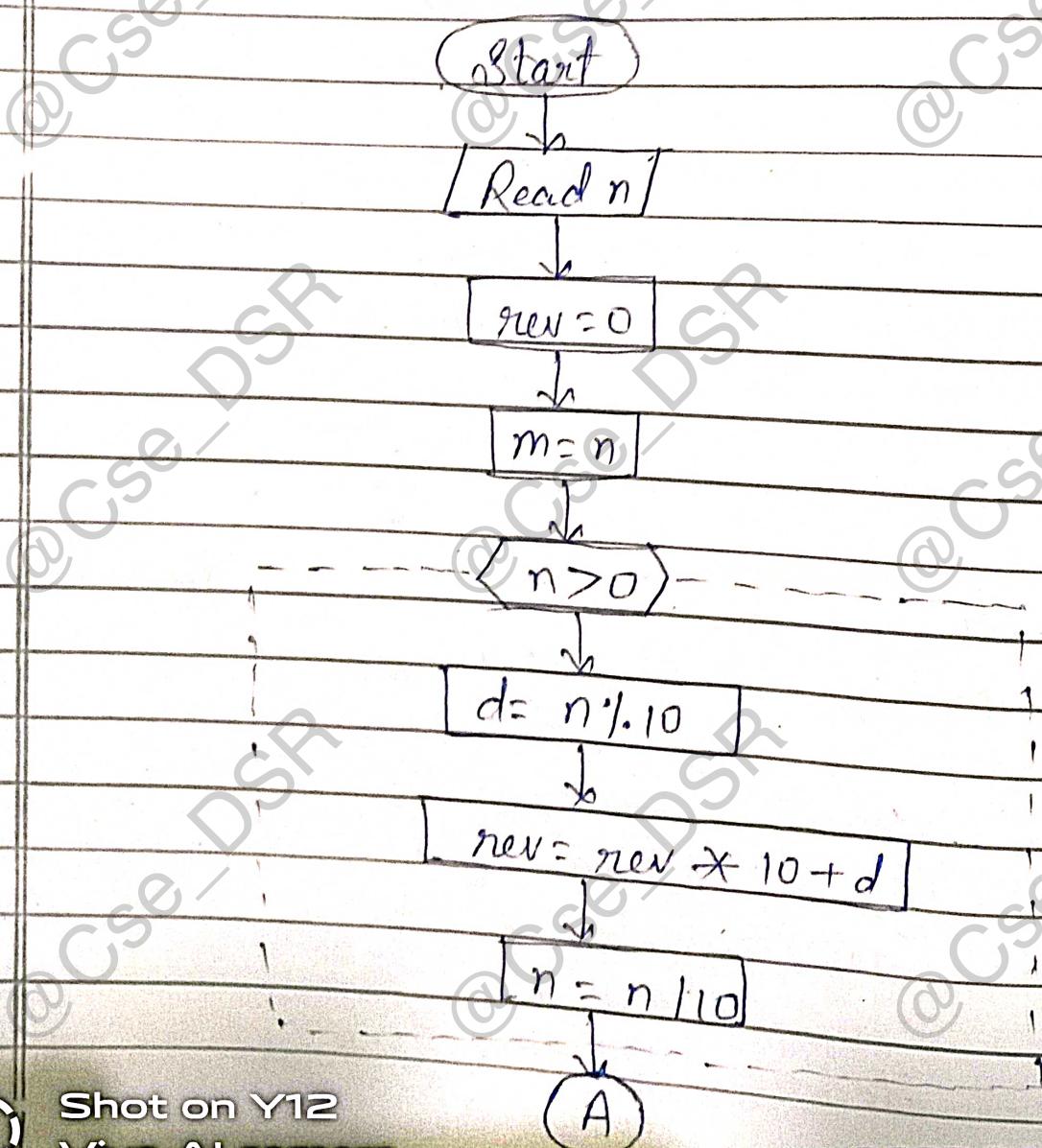


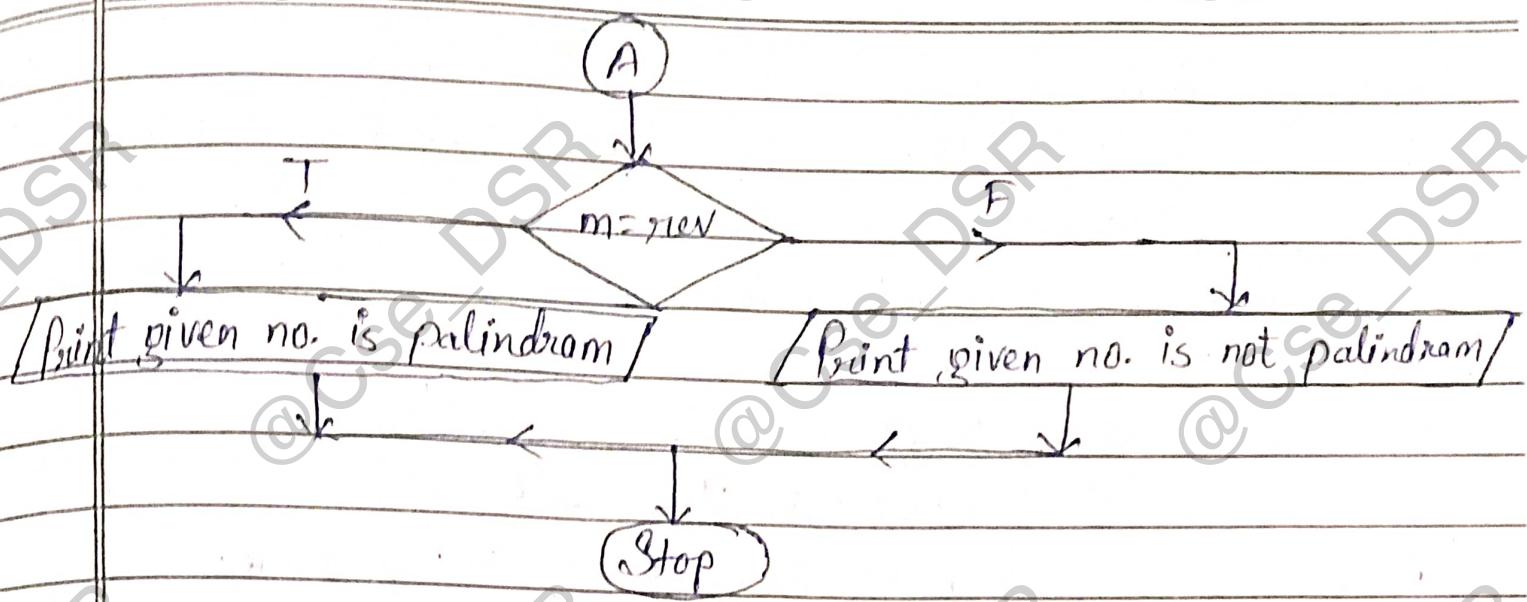
$d = n \% 10$   
 $rev = rev * 10 + d$   
 $n = n / 10$

Step 6: if ( $m = rev$ )  
 print given no. is palindrome  
 else  
 print given no. is not palindrome.

Step 7: Stop

Flowchart:-





Q7. Write an algorithm and draw the flowchart to check given number is Armstrong or not.

Step 1 : Start

Step 2 : Read n

Step 3 : sum = 0

Step 4 : m = n

Step 5 : loop ( $n > 0$ )

$$d = n \% 10$$

$$\text{sum} = \text{sum} + d * d * d$$

$$n = n / 10$$

Step 6 : if ( $m == \text{sum}$ )

print given number is armstrong

else

print given number is not armstrong.

Step 7 : Stop



Dry run :-

$$n = 234$$

$$\text{sum} = 0$$

$$m = 234$$

$$234 > 0$$

$$d = 234 \% 10 = 4$$

$$\text{sum} = \text{sum} + 4 * 4 * 4 = \cancel{64}$$

$$= 0 + 4 * 4 * 4 = 64$$

$$n = 234 / 10 = 23$$

$$23 > 0$$

$$d = 23 \% 10 = 3$$

$$\text{sum} = \text{sum} + 3 * 3 * 3$$

$$= 64 + 27 = 91$$

$$n = 23 / 10 = 2$$

$$2 > 0$$

$$d = 2 \% 10 = 2$$

$$\text{sum} = \text{sum} + 2 * 2 * 2$$

$$= 91 + 8 = 99$$

$$n = 2 / 10 = 0$$

$$0 > 0 \quad \times$$

$$234 = = 99$$

given number is not armstrong



Flowchart:-

