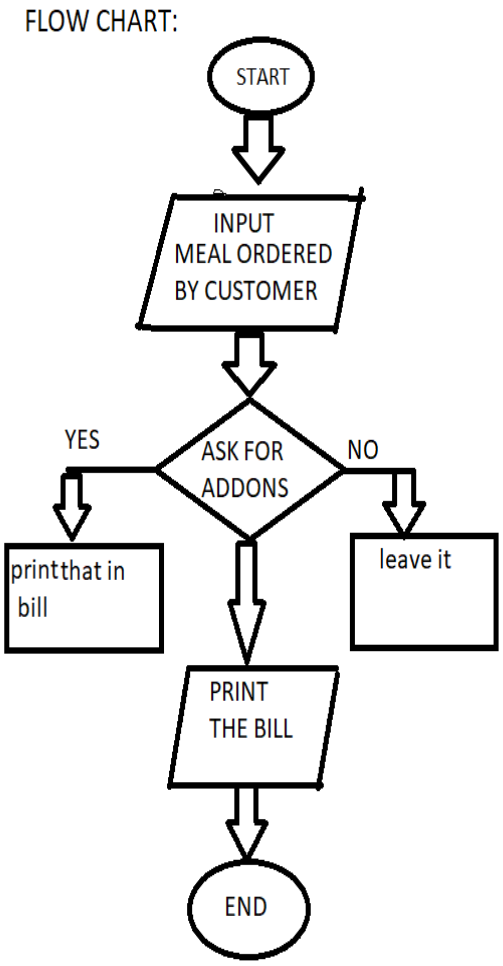
Q1.flowchart



Q1 PSEUDOCODE

01. START

02. INPUT MEAL ORDERED BY THE CUSTOMER

03. IF special add on THEN

    add that in bill amount

04. ELSE

05. leave it

06. Print the bill

07. END

Q1. ALGORITHM:

STEP1: START

STEP2: Take order from customer

STEP3: Ask for add on or special request

STEP4: If there is a special request add that in bill amount

    Else leave it

STEP5: Print the bill

STEP6: END

Q2. Pseudocode:

01. START

02. INPUT check from customer

03. Check the account validity and deposit amount conditions

04. If the entered amount <= balance available in the account then

    Withdraw the amount

ELSE

PRINT “INSUFFICIENT BALANCE”

06. END

Q2. ALGORITHM:

STEP 1: START

STEP 1: INPUT the check from customer

STEP 1: Check the account validity and deposit amount conditions

STEP 1: If the entered amount <= balance available in the account

STEP 1: Withdraw the amount

STEP 1: ELSE

STEP 1: PRINT “INSUFFICIENT BALANCE”

STEP 1: END

Q3. Pseudocode:

```
01. START
02.INPUT A, B, C
03.IF A > B THEN
04.IF A > C THEN
        A is GREATEST
    ELSE
        C is GREATEST
05.ELSE IF B > C THEN
        B is GREATEST
    ELSE
        GREATEST = C
07. PRINT GREATEST
08. END
```

Q3. Algorithm:

STEP 1: **START**

STEP 2: **INPUT** the three numbers: A, B, C

STEP 3: **IF** A is greater than B:

STEP 4: **Then** check if A is greater than C:

If true, A is the greatest

Else, C is the greatest

STEP 5: **ELSE** if B is greater than A

Check if B is greater than C:

If true, B is the greatest

Else, C is the greatest

STEP 6: **Output** the greatest number

STEP 7: **End**

Q4. Algorithm:

STEP1: **START**

STEP2: **Input:**

Ask the user to enter a number between 1 and 12.

STEP3: **VERIFY Input:**

Confirm that the number is within the range of 1 to 12.

STEP4: **Map Number to Month:**

Use a list to map each number to its corresponding month name

STEP:5 **OUTPUT:**

Output the month corresponding to the entered number.

STEP 6: END

Q5 pseudo code:

01. START

02. INPUT two numbers and a one operator (i.e + or -) from user

03. IF the user inputs ‘+’

    Then sum = num1+num2

04. Print the sum

05. Else difference = num1 -num2

06. Print difference

07. END

Q7 ALGORITHM:

STEP 1: START

STEP 2: INPUT Math Problem:

like "5 + 3" or "8 - 2."

STEP 3: Choose What to Do:

Look at the problem and see what math operator you have:

+ for addition, - for subtracting,* for multiplying, / for dividing

and % for finding the remainder

STEP 4: Do the Math:

do the math with the numbers you have:

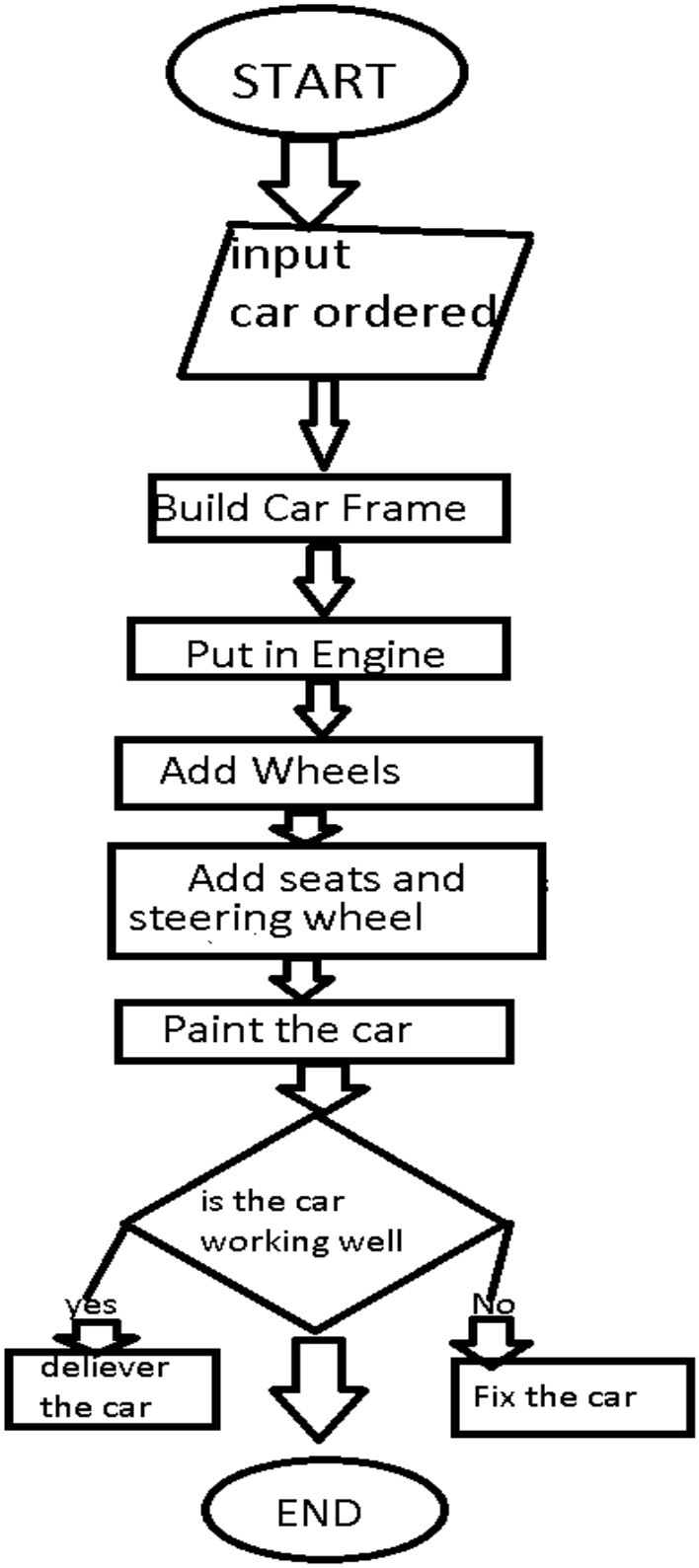
For +, add the numbers. For -, subtract the numbers.

For *, multiply the numbers. For /, divide the numbers. %, find out what's left after dividing.

STEP 5: PRINT the Answer:

STEP 6: END

Q6. FLOW CHART:



Q9: Why we use .gitignore?

ANSWER: We use `.gitignore` file to tell Github which files and folders to not track.

For skipping over things like temporary files or system logs, keeping your project neat and clean.
Only the important files get tracked and shared.
As, we can see from its name ".gitignore" it has word ignore which mean git has to ignore these files.

Q10: Difference between Algorithm and Pseudocode?

Answer:

Algorithm:

A step-by-step procedure for solving a problem.

Should be unambiguous, clear and understandable.

Pseudocode:

A simplified version of programming code written in plain English or a structured form.

Outlines the logic of an algorithm before implementing it in a specific programming language.

More detailed than algorithms but less formal than actual code.

Can be easily translated into various programming languages.