

# **“LAB TASK”**

## **Question no.1:**

Design a flowchart, Pseudocode, Algorithm for processing a customer order at a restaurant, including handling special requests (Like add on).

## **“Pseudo code”**

START

PRINT “WELCOME! How may I help you?”

DISPLAY Menu

PRINT “What would you like to have?”

READ Order

IF Order is available THEN

    PRINT “Do you like to have something else!”

    READ Response

    IF Response = “Yes” THEN

        PRINT “Please enter your special request and add-ons”

        READ SpecialRequest

    ELSE

        SpecialRequest = “None”

    END IF

CALCULATE Bill

PRINT “Here’s your bill”

READ Payment

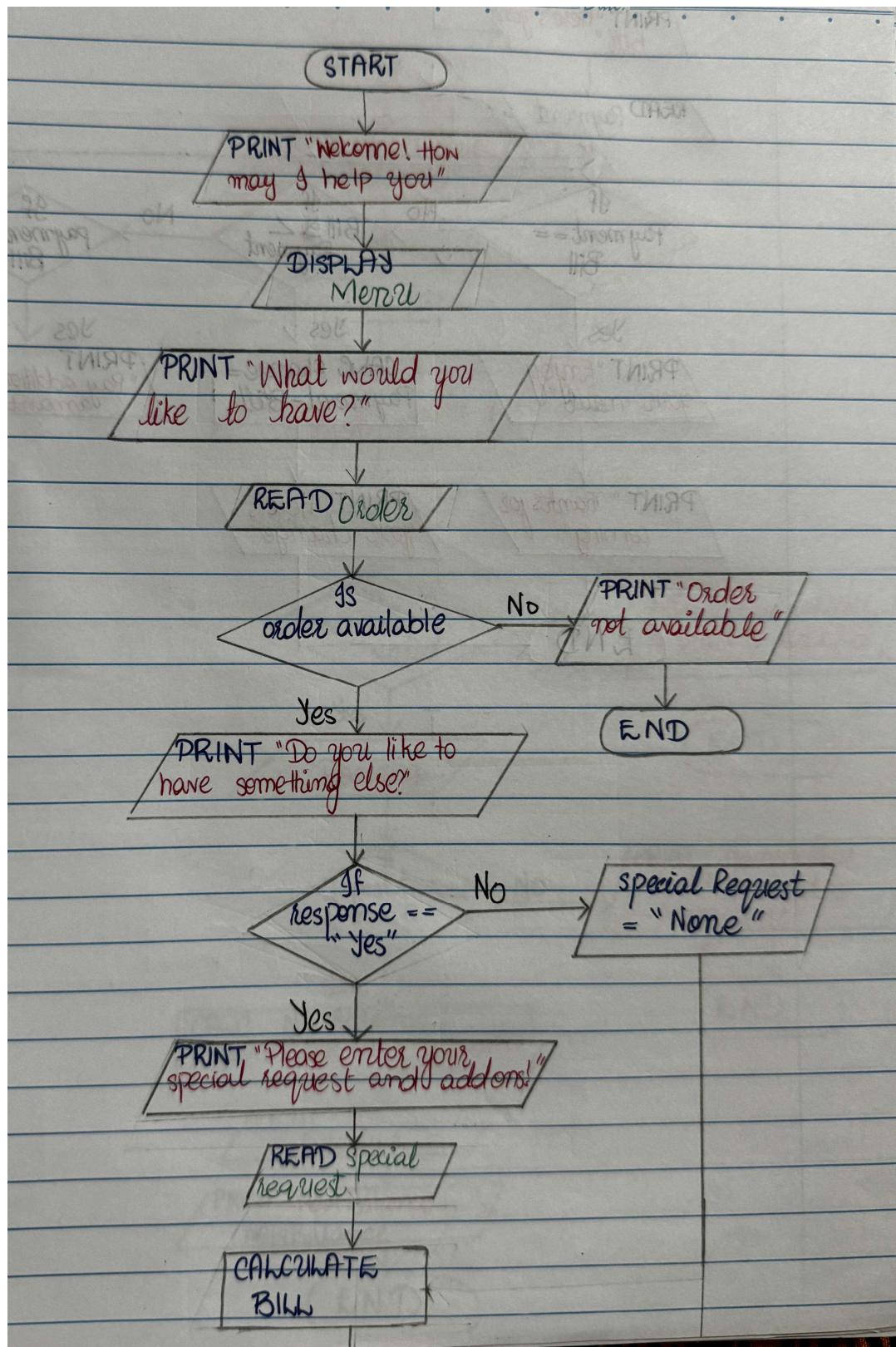
IF Payment == Bill THEN

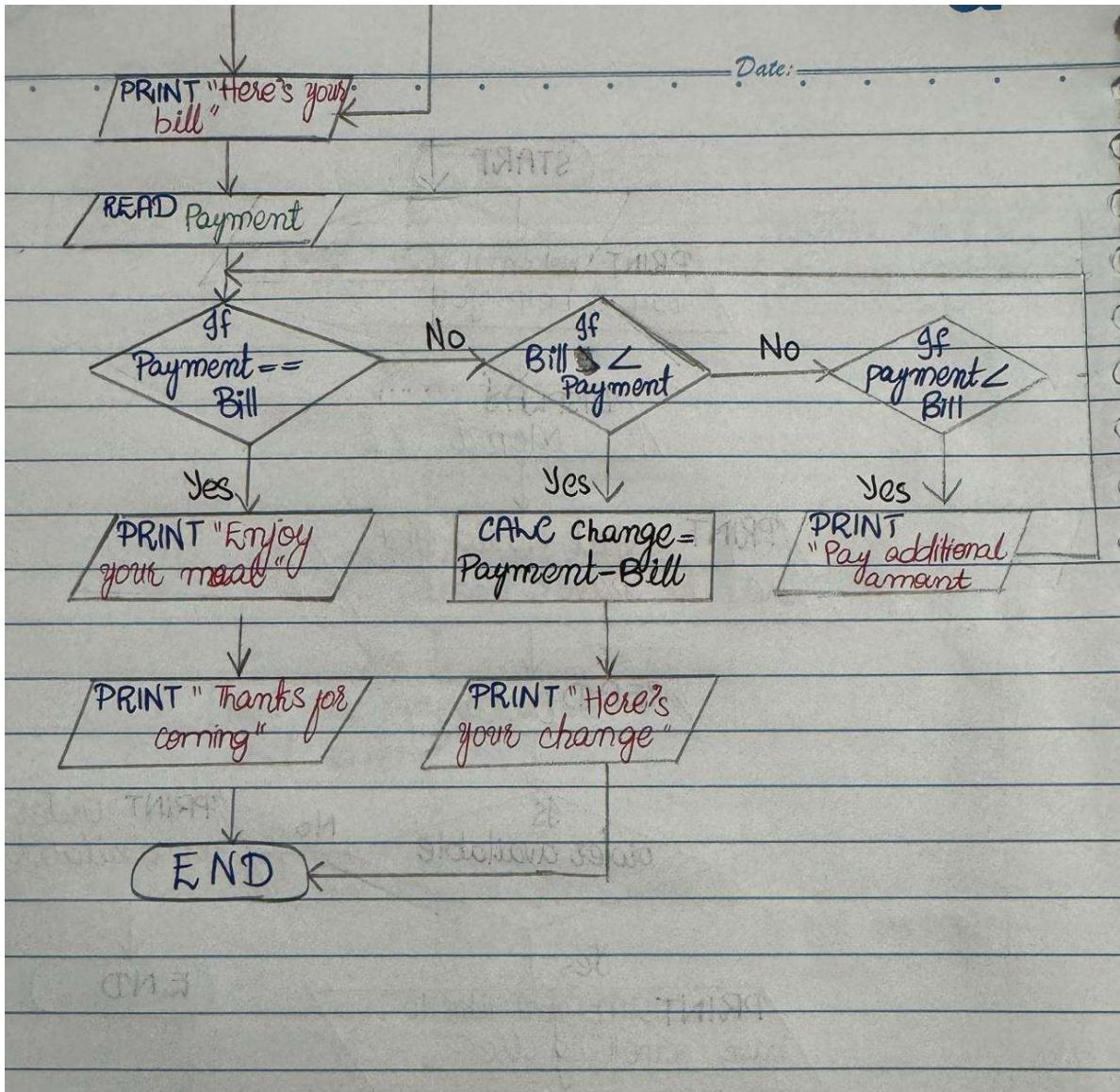
    PRINT “Enjoy your meal!

```
PRINT "Thanks for coming"  
ELSE IF Payment > Bill THEN  
    CALCULATE change = Payment - Bill  
    Print "Here's your change: ",change  
ELSE IF Payment < Bill THEN  
    PRINT "Pay additional amount"  
END IF  
ELSE "Order not available"  
END IF  
END
```

## **“Algorithm”**

- Welcome the customer and display the bill
- Ask the customer what would they like to order
- Check if the order is available
- If yes
- Ask them if they want anything else
- If yes
- Provide the bill to the customer
- If payment is equal to the bill
- Display “Enjoy your meal” and “Thanks for coming”
- Else if payment is greater than the bill
- Calculate the change
- Else if payment is less than the bill
- Display “Pay the additional amount”
- If order is not available
- Display “Order not available”





## **QUESTION NO.2 :**

Design a flowchart, Pseudocode, Algorithm for handling a customer's deposit transaction at a bank, including checks for account validity and deposit amount conditions.

### **“PSEUDO CODE”**

START

PRINT “Enter your current balance”

READ CurrentBalance

PRINT “Enter the amount you want to deposit”

READ DepositAmount

IF account != “Valid” THEN

    PRINT “Transaction failed!! Account is not valid”

ELSE

    PRINT “Processing.....”

    IF DepositeAmount <= 0 THEN

        PRINT “Transaction failed!! Amount must be greater than zero.”

    ELSE

        CALC NewBalance = CurrentBalance + DepositeBalance

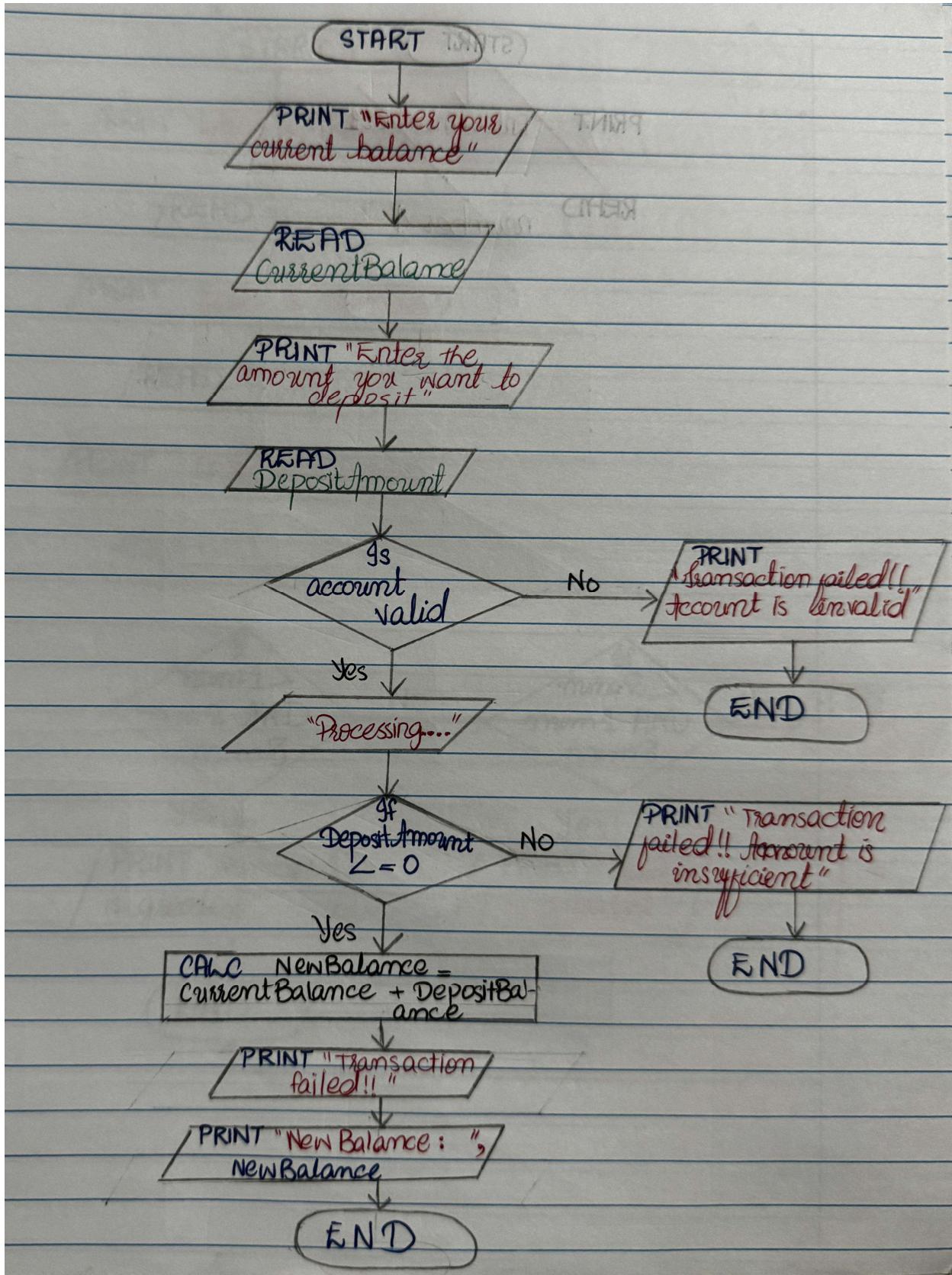
        PRINT “Transaction successful!!”

        PRINT “New Balance:,” NewBalance

    END IF

END IF

END



# **“ALGORITHM”**

- Ask the user to enter the current balance
- Ask the user to enter the deposit amount
- Check if the amount is valid
- If the account is not valid, print a message “Transaction failed!! Account is not valid.
- Else if the account is valid then proceed the further step
- Check if the deposit amount is smaller than or equal to zero throw a message “Transaction failed!! Deposit amount must be greater than zero
- Set newbalance = accountbalance + depositamount
- And display new balance for the user

## **Question no.3:**

Design a flowchart, Pseudocode, Algorithm to determine which of three provided numbers is the greatest.

# **“PSEUDO CODE”**

START

PRINT “Enter the number 1”

READ number 1

PRINT “Enter the number 2”

READ number 2

PRINT “Enter the number 3”

READ number 3

IF number1>number2 AND number1>number3 THEN

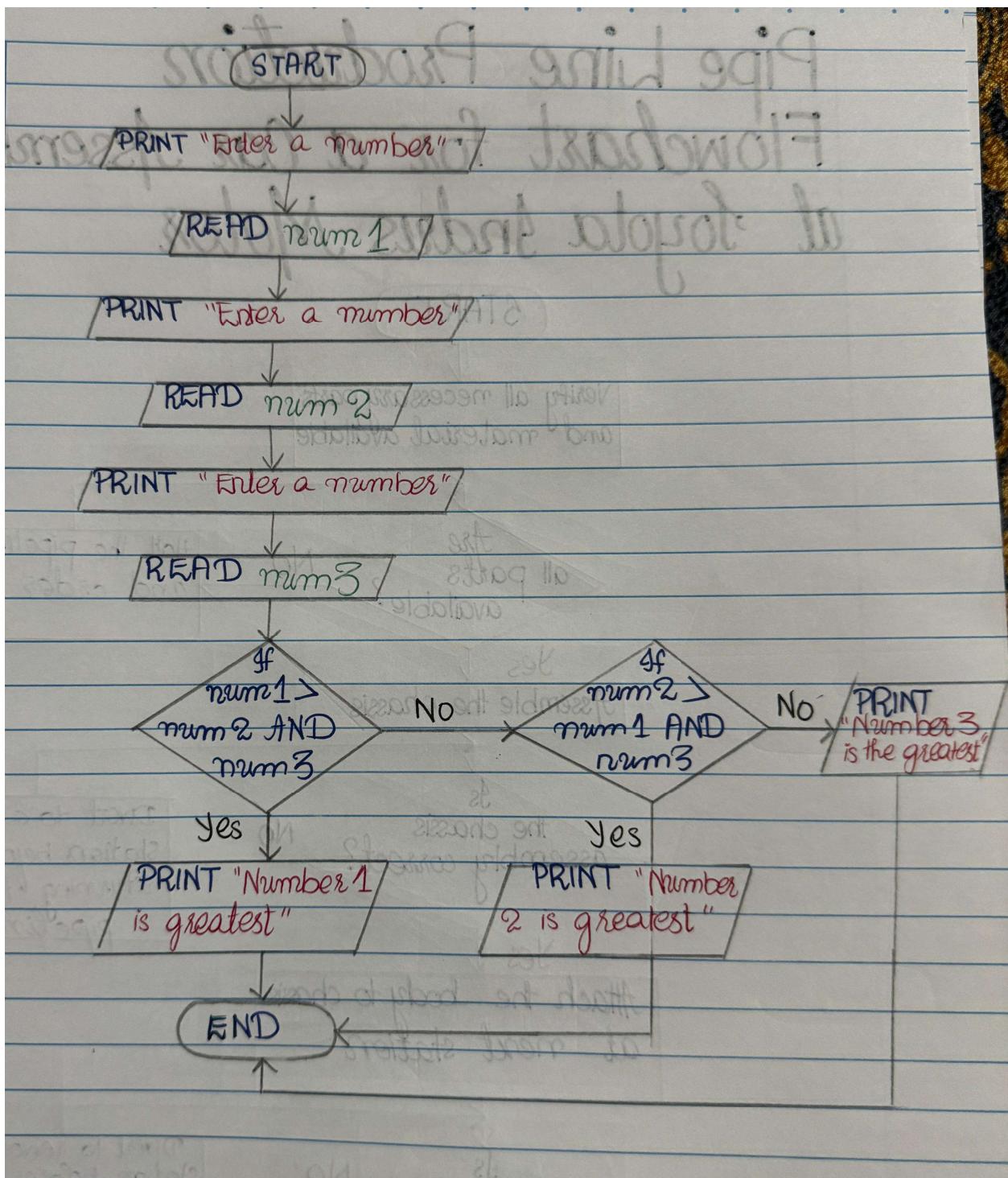
    PRINT “Number1 is greatest”

ELSE IF number2>number3 AND number2>number1 THEN

```
PRINT "Number2 is greatest"  
ELSE  
    PRINT "Number3 is greatest"  
END IF  
END
```

## **“ALGORITHM”**

- Ask the user to enter three numbers named as
- “number1”
- ”number2”
- “number3”
- Check if “number 1” is greater than both “number 2” and “number 3” then
- Number1 is the greatest.
- Check if “number 2” is greater than “number 1” and “number 3”
- then
- Number2 is greatest.
- If “number 1” neither “number 2” will be greater
- then
- Number3 will be greatest.



## **QUESTION NO.4:**

Implement an algorithm where the user enters a number, and an appropriate month is displayed.

### **“ALGORITHM”**

- Ask the user to enter a number between 1-12
- If the number is 1, display “JANUARY”.
- If the number is 2, display “FEBUARY”.
- If the number is 3, display “MARCH”.
- If the number is 4, display “APRIL”.
- If the number is 5, display “MAY”.
- If the number is 6, display “JUNE”.
- If the number is 7, display “JULY”.
- If the number is 8, display “AUGUST”.
- If the number is 9, display “SEPTEMBER”.
- If the number is 10, display “OCTOBER”.
- If the number is 11, display “NOVEMBER”.
- If the number is 12, display “DECEMBER”.

## **QUESTION NO.5 :**

Create pseudocode a small calculator which only does ‘+’ or ‘-’ Operations. (Hint: Take three Variable inputs with one being used for the operator)

### **“PSEUDO CODE”**

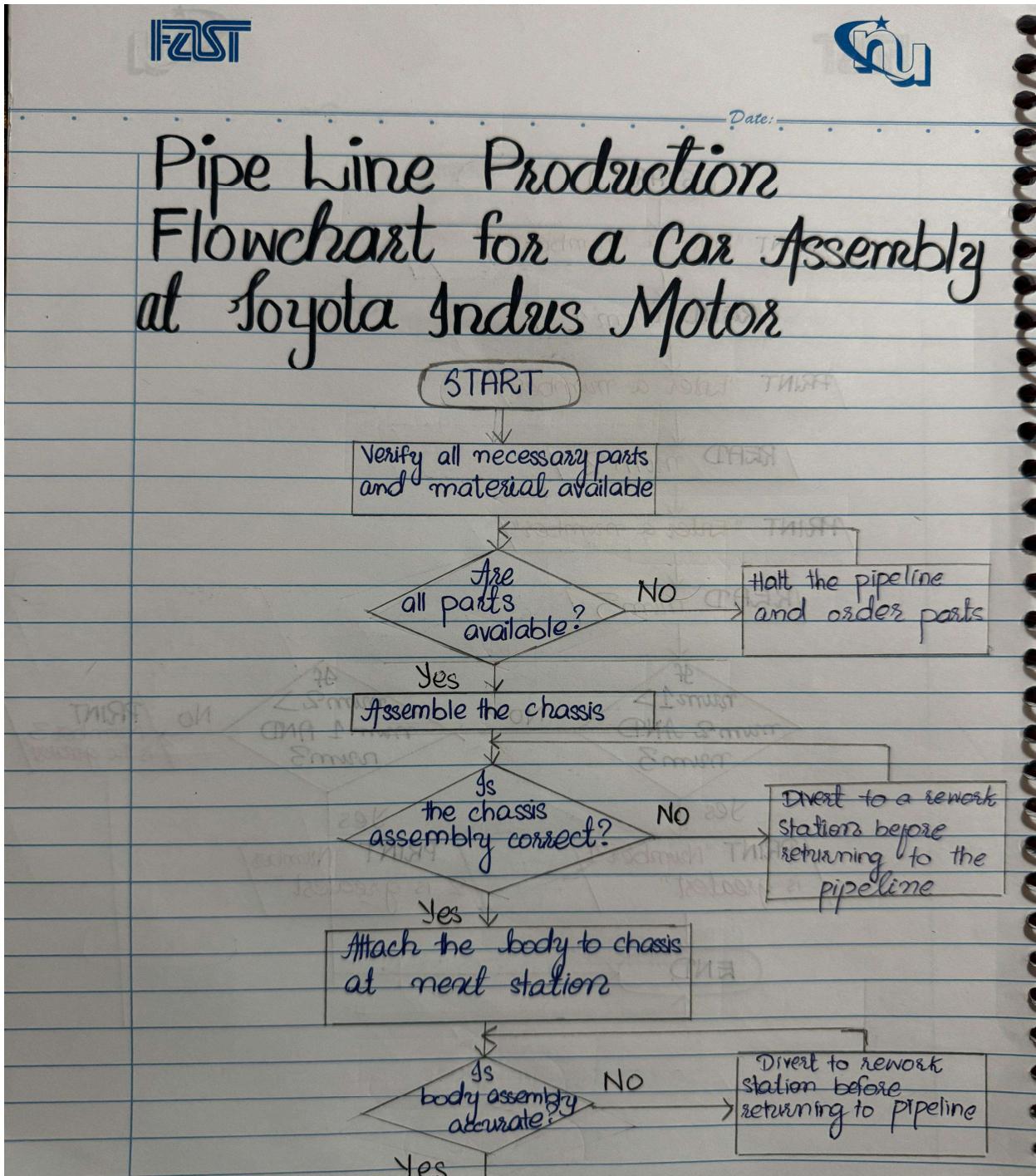
START

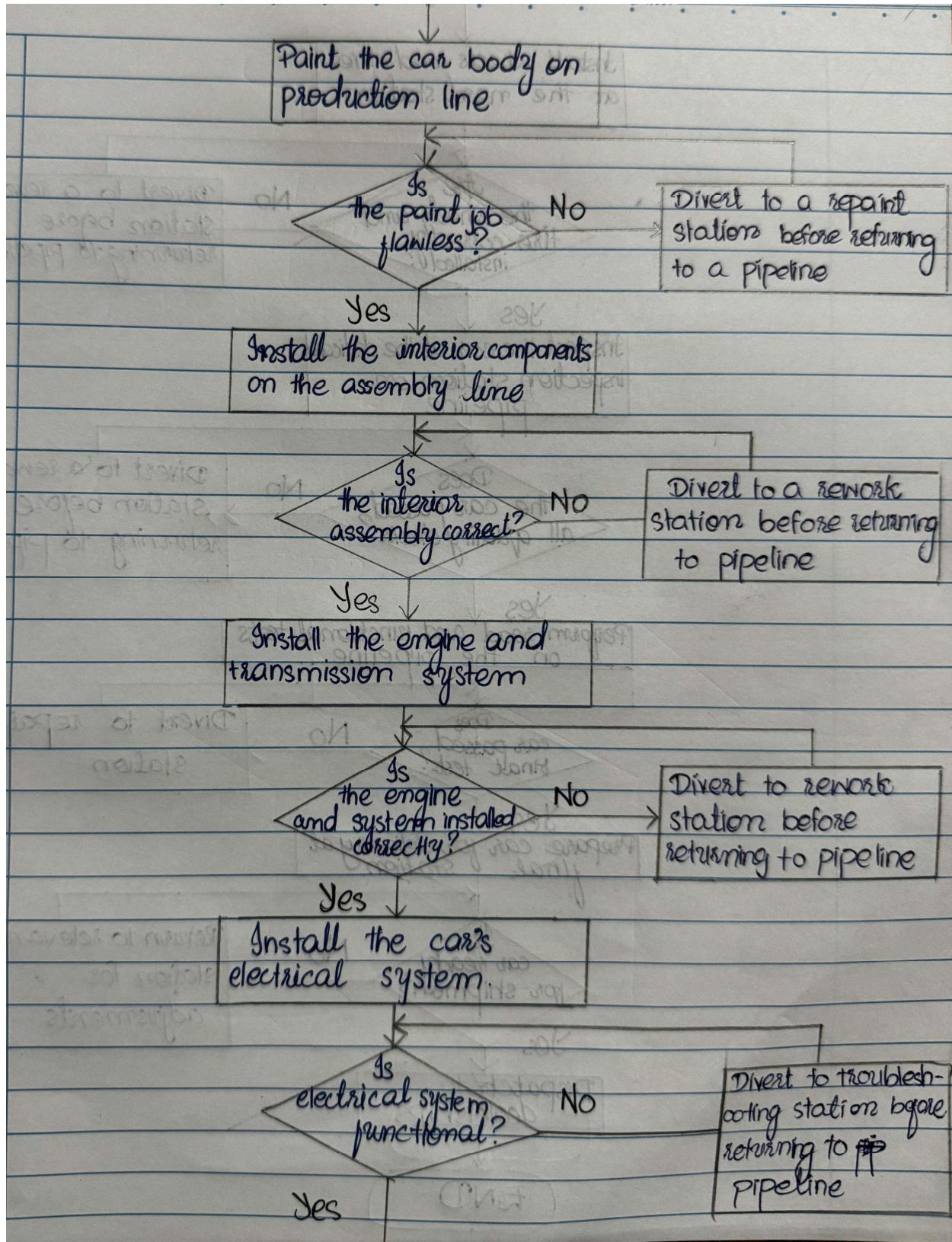
PRINT “Enter number 1”

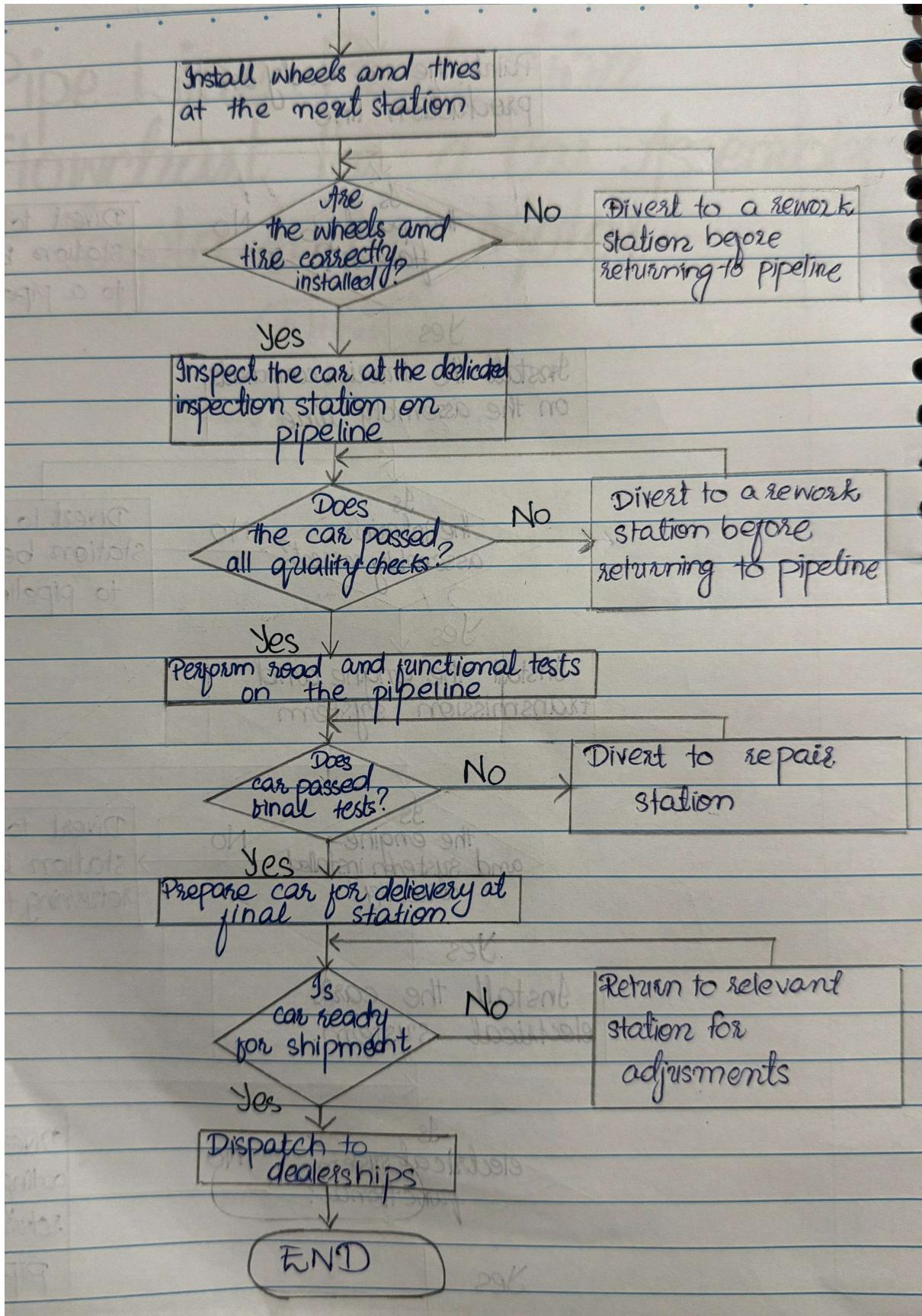
READ number 1

```
PRINT "Enter number 2"
READ number 2
PRINT "Enter number 3"
READ number 3
PRINT "Which operation do you want to perform? (+ OR -)"
READ operation
IF operation == "+" THEN
    CALCULATE Result = number1 + number2 + number3
    PRINT "The sum is ", Result
ELSE IF operation == "-" THEN
    CALCULATE Result = number1 - number2 - number3
    PRINT "The difference is ", Result
ELSE
    PRINT "Invalid Operation Selected"
END IF
END
```

## **QUESTION NO.6:**







## **QUESTION NO.7:**

Implement an algorithm for making a simple calculator with all the operators (+,-,\*,/,%)

### **“ALGORITHM”**

- Ask the user to input a number1
- Ask the user to input a number2
- Ask the user to select any operation
- If the operation is “+”
- Calculate Result = number1 + number 2
- Display Result
- If the operation is “-“
- Calculate Result = number1 – number2
- Display Result
- Else if operation is “ \* ”
- Calculate Result = number1 \* number2
- Display Result
- Else if operation is “ / ”
- Check number 2 is not zero
- Calculate Result = number1/number2
- Display Result
- Else
- Display “Error: Division by zero is not allowed”
- Else if operation is “%”
- Check number 2 is not zero
- Calculate Result = number%number2
- Else
- Display “Error: Division by zero is not allowed”
- Else
- Display “Invalid Operation”

## **QUESTION NO.9:**

Why do we need to use .gitignore ?

- To exclude unnecessary files
- To reduce repository size
- To maintain clean history
- To avoid including user-specific or environment specific files that could cause issues for others

## **QUESTION NO.10:**

Difference between pseudocode and algorithm ?

An algorithm is a step-by-step procedure to solve a problem, while pseudocode is a simplified, human readable representation of that algorithm.

Pseudocode helps in planning the algorithm in a language-agnostic way, making it easier to convert it into actual code later

