

LAB TASKS

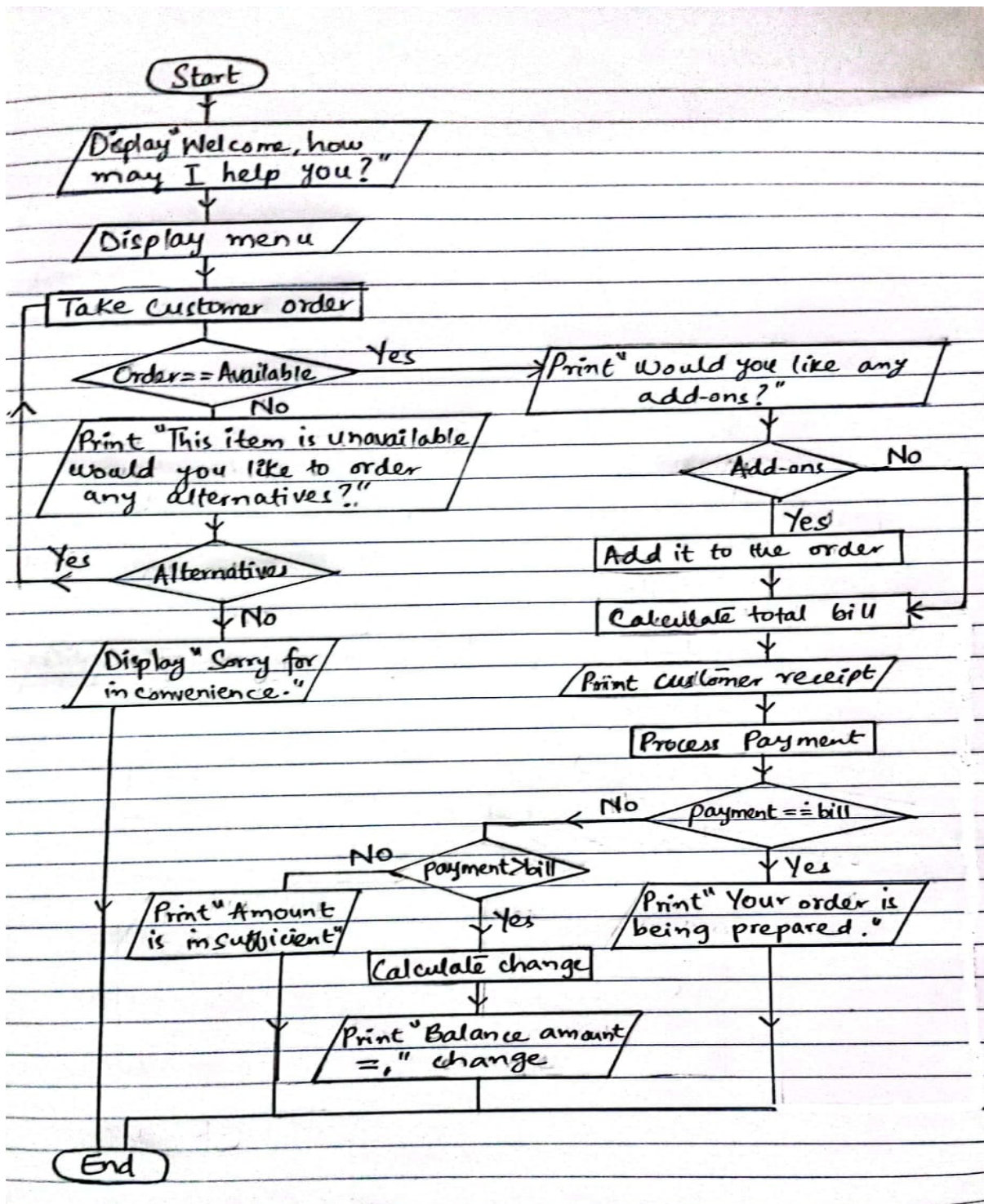
LAB#2

Q1. Design a flowchart, Pseudocode, Algorithm for processing a customer order at a restaurant, including handling special requests (Like add on).

Algorithm:

1. Start.
2. Greet the customer.
3. Display menu.
4. Take the customer's order.
5. Check if item is available ask if customer has any special requests.
 - If yes, add special requests to the order.
 - If no, proceed to the next step.
6. If any item is unavailable
 - Inform the customer that the item is not available.
 - Ask if the customer wants to order an alternative.
 - If yes, take the new order and check if the new item is available.
 - If item is not available repeat until an available item is selected.
 - If the customer does not want to order an alternative
 - End the process with a sorry note.
7. Calculate the total bill and print customer receipt.
8. Ask for payment.
9. Give the change if applicable or ask for more amount if the payment is insufficient.
10. Tell the customer their order will be ready in 20 minutes.
11. End.

Flowchart:



Pseudo-Code:

1. Start
2. Display "Welcome, how may I help you?"
3. Display "Menu"
4. Take customer order.
5. Read order.
6. IF item is available THEN,
 Print "Would you like any add-ons?"
 IF yes, THEN
 Add it to the order.
ELSE
 Print "This item is not available. Would you like to order any alternatives?"
 IF yes, THEN
 Repeat STEP 5.
ELSE
 Print "Sorry for your inconvenience."
7. Calculate the total bill.
8. Read total bill.
9. Print customer receipt.
10. Get payment.
11. If payment==bill
 Display "Your order is being prepared"
Else if payment> bill
 Calculate change amount
 Display "Balance amount=," change amount
12. Else
 Display "Amount is insufficient"
13. End.

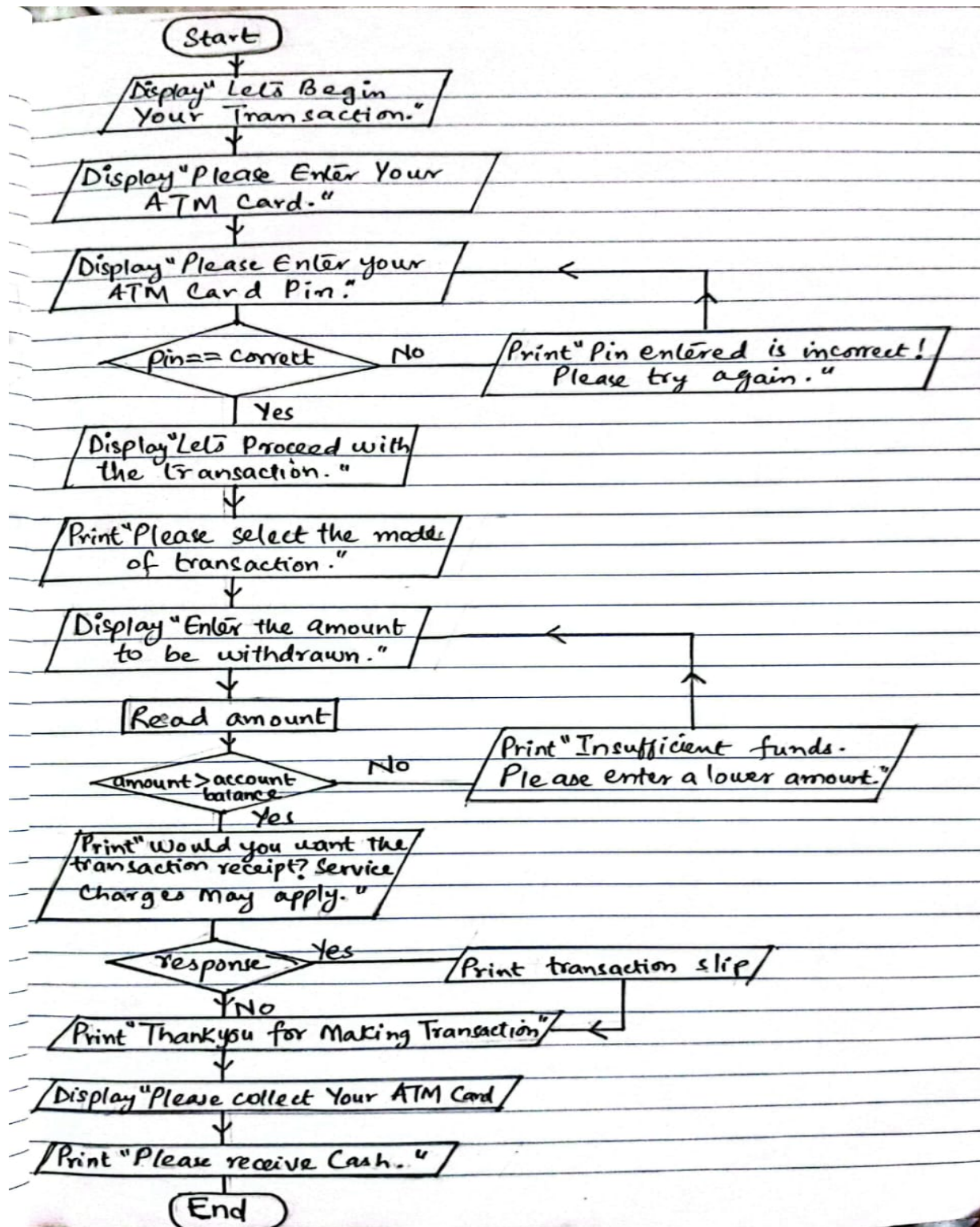
Q2. Design a flowchart, Pseudocode, Algorithm for handling a customer's deposit transaction at a bank, including checks for account validity and deposit amount conditions.

Algorithm:

1. Start.
2. Greet the user.
3. Ask the user to insert their ATM Card.
4. Ask the user to enter card pin.

5. If pin entered is correct ask the mode of transaction whether it is bank transfer, cash withdrawal etc. in case the pin is incorrect ask the user to try again until 3 attempts if the pin is still incorrect block the card and notify the bank.
6. Ask to enter the amount which has to be withdrawn.
7. Check if the entered amount is available in the user's account.
8. If funds are insufficient,
inform the user and allow them to enter a lower amount or cancel the transaction.
9. Ask if the customer would want the transaction receipt or not.
10. Return the card.
11. Withdraw the selected amount of cash.
12. In case of any technical issue notify the user and provide instructions for resolving the issue.
13. Display a message thanking them and reminding them to receive their card and cash.
14. End.

Flowchart:



Pseudocode:

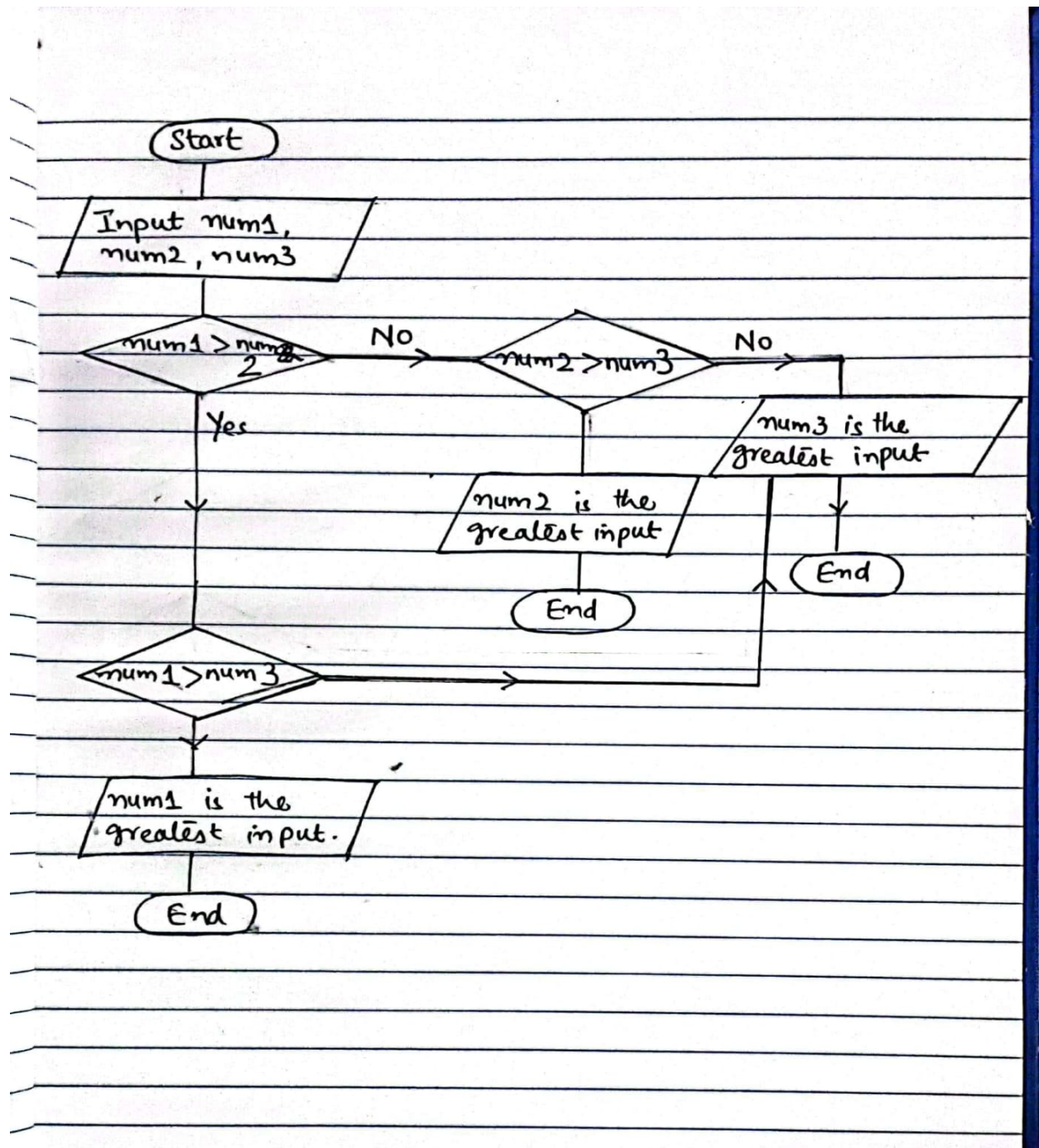
1. Start.
2. Set mode of transaction ==Cash withdrawal.
3. Display "Let's Begin Your Transaction."
4. Display "Please enter your ATM Card."
5. Display "Please enter your ATM Card pin."
6. Read ATM Card pin.
7. If pin==correct
 Display "Lets proceed with the transaction."
8. Else
 Display "The pin you entered is incorrect! Please try again."
 Repeat STEP 4.
9. Display "Please select the mode of transaction."
10. Read mode of transaction.
11. Display "Enter the amount to be withdrawn."
12. Read the amount.
13. IF amount > account balance THEN
 Print "Insufficient funds. Please enter a lower amount."
 Repeat STEP 10.
14. ELSE
 Display "Would you want the transaction receipt? Service charges may apply."
 Read response.
 IF response==yes
 Print transaction receipt.
 Display "Thank you for making transaction."
 ELSE
 Display "Thank you for making transaction."
15. Display "Please collect your ATM Card."
16. Display "Please receive the cash withdrawn."
17. End.

Q3. Design a flowchart, Pseudocode, Algorithm to determine which of three provided numbers is the greatest.

Algorithm:

1. Start.
2. Input three numbers A, B, C.
3. Compare among three numbers if A > B and A > C "A is the greatest number."
 A < B and B > C "B is the greatest number."
 B < C and A < C "C is the greatest number."
4. End.

Flowchart:



CS Scanned with CamScanner

Pseudocode:

1. Start.
2. Set num1, num2, num3.
3. Print "Please provide three input numbers."

4. Print "Enter num1 =."
5. Read num1.
6. Print "Enter num2 =."
7. Read num2.
8. Print "Enter num3=."
9. Read num3.
10. IF num1 > num2
 - IF num1 > num3.
 - Print "num1 is the greatest input."
 - ELSE
 - Print "num3 is the greatest."
- ELSE
- IF num2 > num3
- Print "num2 is the greatest input."
- ELSE
- Print "num3 is the greatest input."
11. End.

Q4. Implement an algorithm where the user enters a number, and an appropriate month is displayed.

Algorithm:

1. Start.
2. Ask the user to enter number between 1 and 12(including 1 and 12).
3. Read the user's input.
4. Check if the number is entered correctly, Display
 - 1=January
 - 2=February
 - 3= March
 - 4=April
 - 5=May
 - 6=June
 - 7=July
 - 8=August
 - 9=September
 - 10=October
 - 11=November
 - 12= December

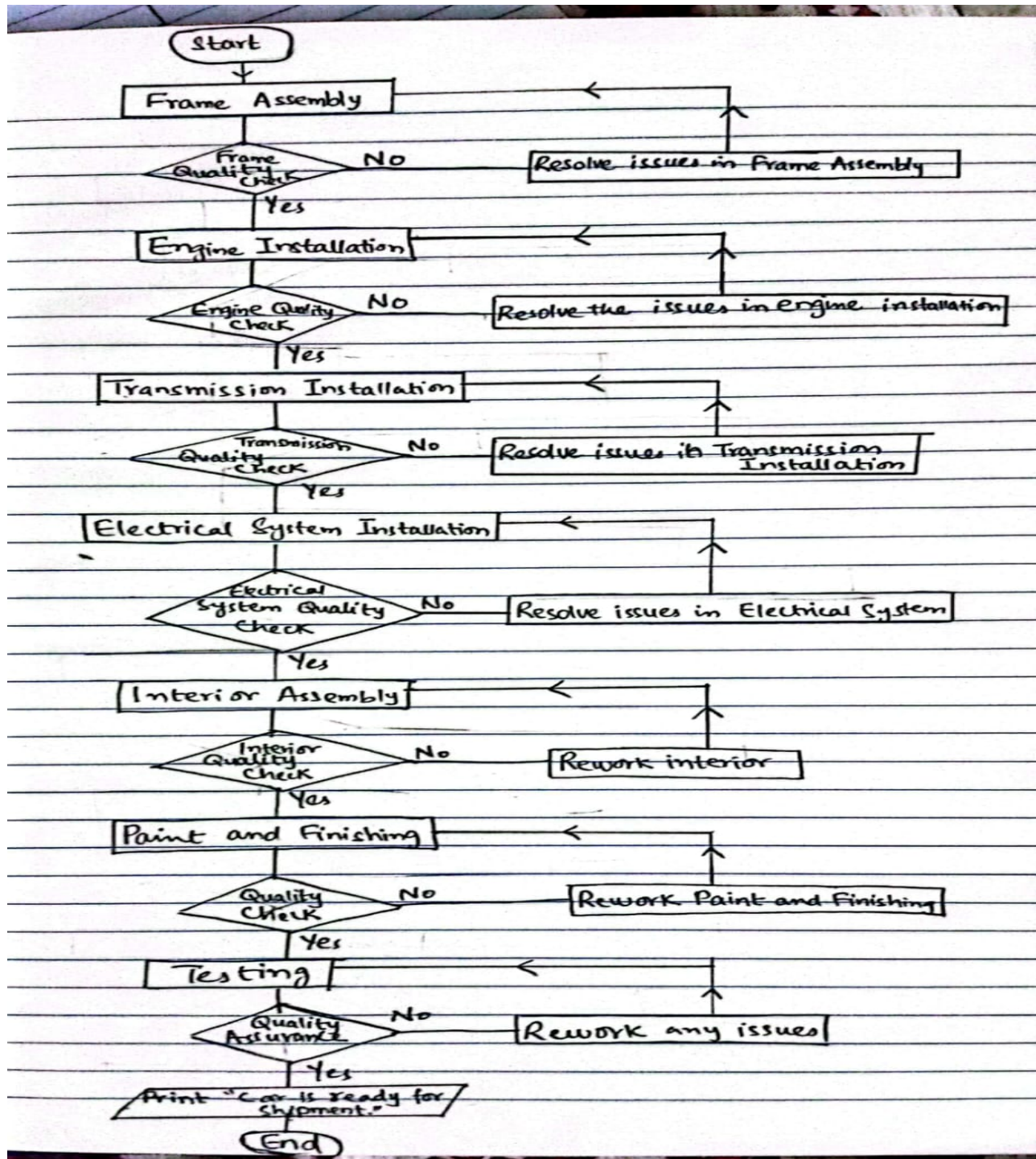
If the number entered is incorrect again ask the user to enter any number between 1 and 12.
5. End.

Q5. Create pseudocode a small calculator which only does '+' or '-' Operations. (Hint: Take three variable inputs with one being used for the operator)

Pseudocode:

1. Start.
2. SET num1, num2.
3. Print "Enter num1="
4. Read num1.
5. Print "Enter the operator '+' or '-'."
6. Read operator.
7. Print "Enter num2="
8. Read num2.
9. IF operator == +
 Result = num1 + num2
10. ELSE IF operator == -
 Result = num1 - num2
11. ELSE
 Print "Invalid input."
12. Print "Result="
13. End.

Q6. You are working at Toyota Indus Motors and want to assemble a car. Design a flowchart with proper process modules and decision structures to replicate a pipeline production.



Q7. Implement an algorithm for making a simple calculator with all the operators (+, -, *, /, %)

Algorithm:

1. Start.
2. Ask the user to enter num1, operator (+, -, *, /, %), num2.
3. Check entered operator
 - for +, add num1 and num2.
 - for -, subtract num2 from num1.
 - for *, multiply num1 and num2.
 - for /, check if num2 is not zero
 - If num2 is not zero, divide num1 by num2.
 - If num2 is zero, show an error message.
 - for %, check if num2 is not zero
 - If num2 is not zero, find num1 % num2.
 - If num2 is zero, show an error message.
4. If any other operator is entered, show an error message.
5. Print the result of the operation performed.
6. End.

Q9. Why we use. gitignore?

ANS: The. gitignore file is a powerful tool that helps developers maintain a clean, efficient, and secure codebase. By ensuring that unnecessary, sensitive, or environment-specific files are not tracked by Git, you create a more streamlined and professional development workflow. This not only makes collaboration easier but also ensures that your project repository remains focused and relevant, containing only the files that are truly necessary for the development and deployment of your project. It is mainly used to Prevent Repository Clutter, Protect Sensitive Information, Avoid Merge Conflicts, Improve Build and Deployment Processes and Maintain Repository Size.

Q10. Difference between Algorithm and Pseudocode?

ANS:

ALGORITHM	PSEUDOCODE
An algorithm is a clear set of instructions needed to be followed in order to solve a problem or complete any task	Pseudocode is a simpler and direct form of an algorithm; it is written in the format of a code but is not meant to be executed.
It is written in plain language and focuses on the sequential steps needed to achieve a result without the concern of any formality in the language.	It uses a blend of plain language and programming like structure but is not related to any other programming language's syntax.
It provides a rough overview how any problem can be solved and different ways to solve a particular problem.	It is a refined form of an algorithm but not actually a code. It basically makes it easier to transform a logic into a program.