

PF Lab Task 2

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

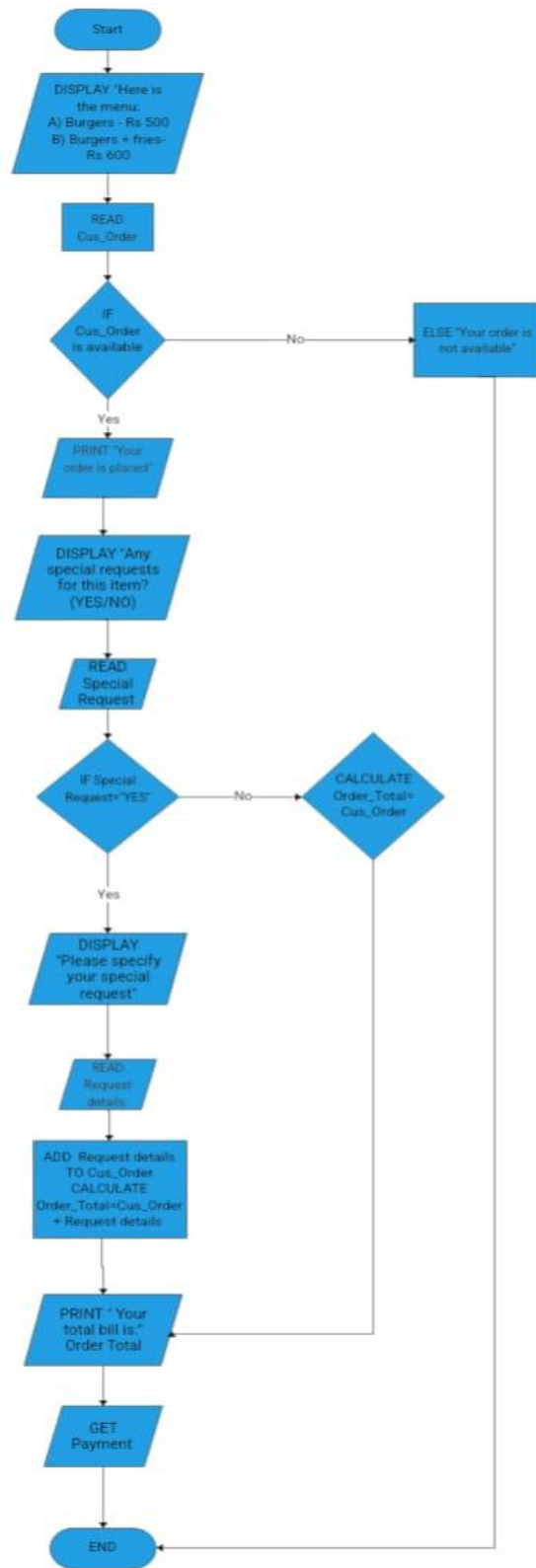
Pseudocode

1. START
2. DISPLAY "Here is the menu:
3. A) Burgers – Rs 500
4. B) Burgers + Fries – Rs 600"
5. READ Cus_Order
6. IF Cus_Order is available THEN
- 7.
8. PRINT "Your order is placed"
9. DISPLAY "Any special requests for this item? (Yes/No)"
10. READ Special Request
11. IF Special Request = "Yes" THEN
12. DISPLAY "Please specify your special request"
13. READ Request details
14. ADD Request details TO Cust_Order
15. CALCULATE Order_Total = Cust_Order + Request details
16. ELSE
17. CALCULATE Order_Total = Cust_Order
18. END IF
19. PRINT "Your total bill is:" Order_Total
20. GET Payment
21. ELSE "Your order is not available"
22. END IF
23. END

Algorithm

1. Show the available menu items.
2. Take the customer's order.
3. Determine if the chosen item is available.
4. Confirm that the order has been placed.
5. Ask if there are any special requests, if so, handle them and adjust the total cost.
6. Display the total amount.
7. Obtain payment details.

Flowchart



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.

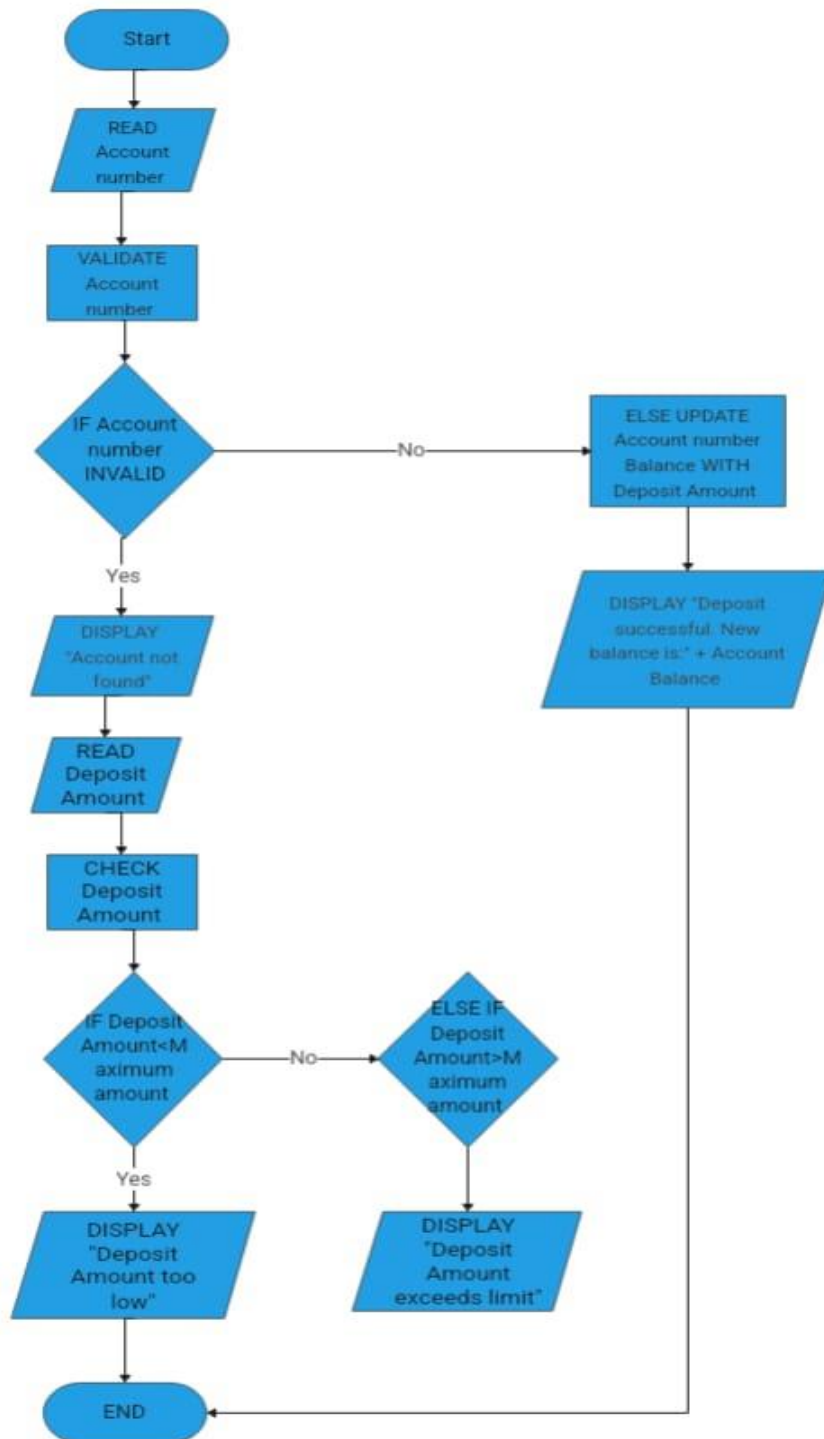
Pseudocode

1. START
2. READ Account Number
3. VALIDATE Account Number
4. IF Account Number INVALID THEN
5. DISPLAY "Account not found"
6. END
7. END IF
8. READ Deposit Amount
9. Check Deposit Amount
10. IF Deposit Amount < Maximum Amount THEN
11. DISPLAY "Deposit Amount too low"
12. END
13. ELSE IF Deposit Amount > Maximum Amount THEN
14. DISPLAY "Deposit Amount exceeds limit"
15. END
16. ELSE
17. UPDATE Account Balance WITH Deposit Amount
18. DISPLAY "Deposit successful. New balance is:" + Account Balance
19. END IF
20. END

Algorithm

1. Read account number from customer input
2. Validate account number
3. If account number is not valid, display "amount not found and terminate the process.
4. If account number is valid proceed to next step
5. Read deposit amount from customer input
6. Check deposit amount:
7. If deposit amount is less than the minimum allowed, display deposit amount too low and terminate the process
8. If deposit amount is greater than the maximum allowed, display deposit amount exceeds limit and terminate the process
9. If deposit amount is valid proceed to the next step
10. Update the account balance with the deposit amount
11. Display "Deposit successful. New Balance is:" account balance

Flowchart



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

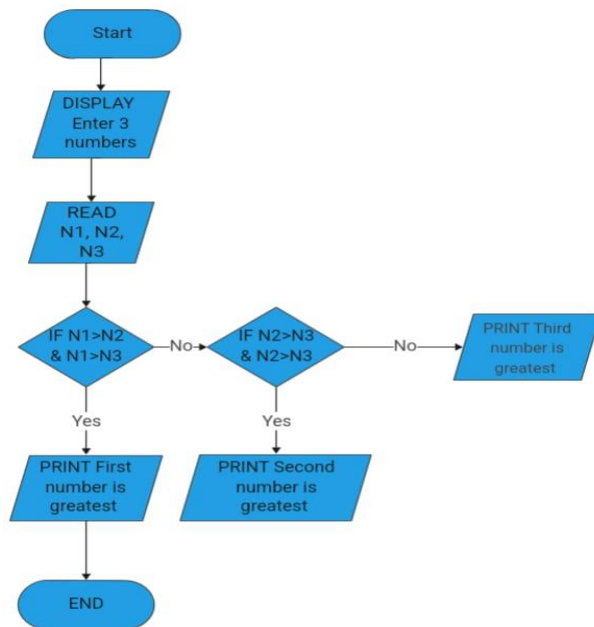
Pseudocode

1. START
2. DISPLAY "Enter 3 Numbers:"
3. READ N1, N2, N3
4. IF (N1>N2, N1>N3) THEN
5. PRINT "First Number is greatest"
6. ELSE IF (N2>N1, N2>N3) THEN
7. PRINT "Second Number is greatest"
8. ELSE
9. PRINT "Third Number is greatest"
10. END IF
11. END

Algorithm

1. Ask the user to enter three numbers
2. Read the numbers
3. If first number is greater than second and third number, then display first number is greatest
4. If second number is greater than first and third number, then display number is greatest.
5. If third number is greater than second and first number then, display third number is greatest.

Flowchart



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

Algorithm

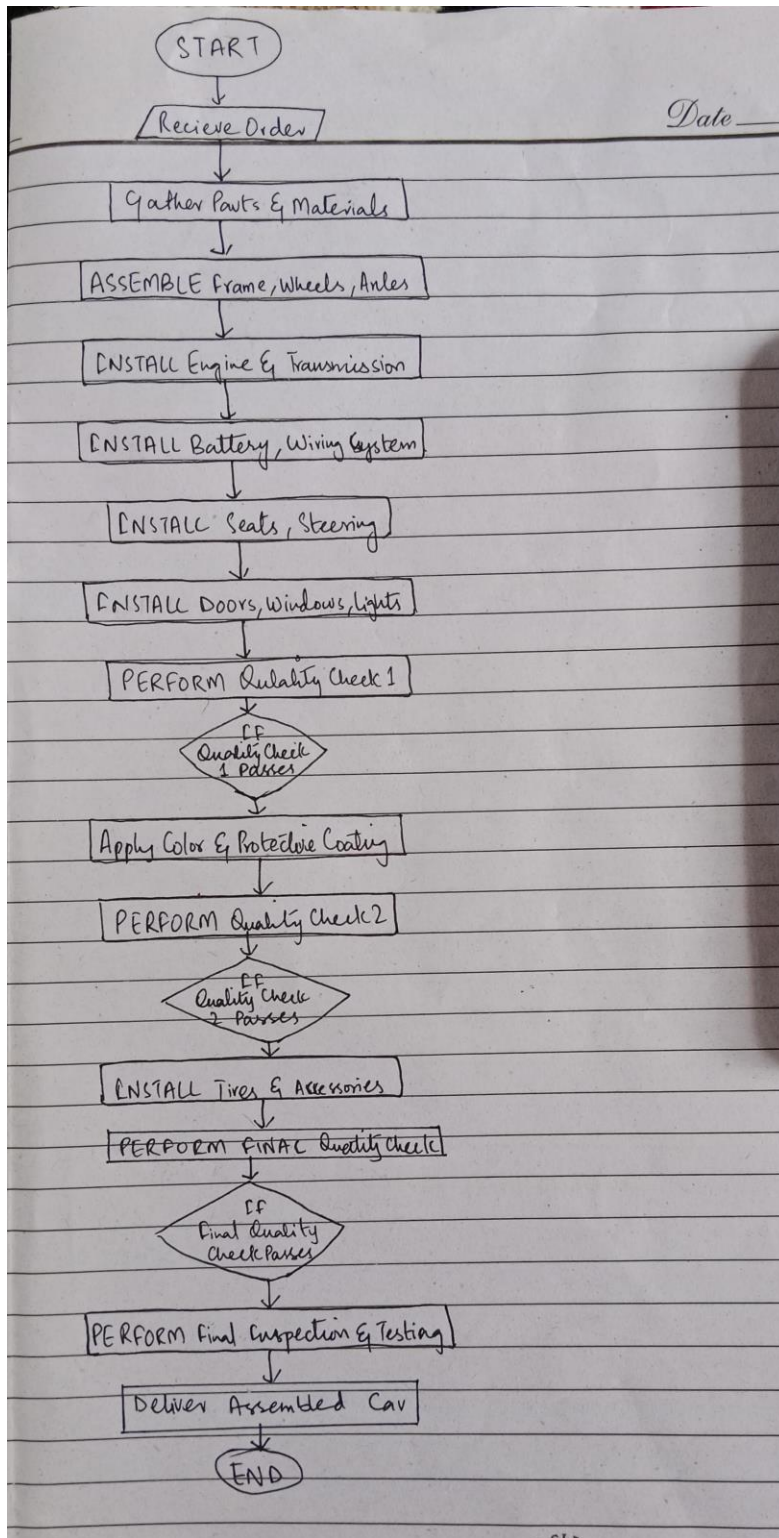
1. Ask the user to enter a number between 1 and 12
2. Check
 - If the number is 1, display January
 - If the number is 2, display February
 - If the number is 3, display March
 - If the number is 4, display April
 - If the number is 5, display May
 - If the number is 6, display June
 - If the number 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
3. If the number is not between 1 and 12, display INVALID NUMBER
4. 5. Create pseudocode of a small calculator which only does '+' or '-' Operations.
(Hint: Take three variable inputs with one being used for the operator)

Pseudocode

1. START
2. DISPLAY "Enter first number"
3. READ N1
4. DISPLAY "Enter the operator + or -"
5. READ op
6. DISPLAY "Enter second number"
7. READ N2
8. IF op = + THEN
9. CALCULATE Sum = N1 + N2
10. PRINT "N1 + N2 =", Sum
11. ELSE IF op = - THEN
12. CALCULATE Difference = N1 - N2
13. PRINT "N1 - N2 =", Difference
14. ELSE
15. PRINT "INVALID OPERATOR"
16. END IF
17. END

6. 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.

Flowchart



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

Algorithm

1. Ask the user to enter first number
2. Read the number
3. Ask the user to enter an operator (+, -, *, /, %).
4. Read the operator
5. Ask the user to enter second number
6. Read the number
7. If the operator is +, add the two numbers.
8. If the operator is -, subtract the second number from the first.
9. If the operator is *, multiply the two numbers.
10. If the operator is /, divide the first number by the second.
11. If the operator is %, find the modulus when the first number is divided by the second.
12. Display the result of the operation.
13. END

9. Why we use .gitignore?

The .gitignore file is used to get to specify which files and directory should be ignored by git when tracking changes in the repository. This means that any file or directory listed in the .gitignore file will not be added to the repository or included in commits even if it exists in your local working directory.

10. Difference between Algorithm and Pseudocode?

Algorithm	Pseudocode
An unambiguous specification of how to solve a problem.	An informal high-level description of the operating principle of a computer program or other algorithm.
Helps to simplify and understand the problem.	A method of developing an algorithm.