PF LAB 02

TASK 1: PROCESSING A CUSTOMER'S ORDER AT A RESTAURANT

ALGORITHM:

- START
- HAVE THE WAITER TAKE THE ORDER
- ASK THE CUSTOMER IF HE HAS ANY SPECIAL ADD ONS/ REQUEST FOR HIS ORDER
- PASS THE ORDER TO THE CHEF
- WAIT FOR THE ORDER TO COOK
- SERVE THE ORDER TO THE CUSTOMER
- PRESENT THE BILL TO THE CUSTOMER
- COLLECT THE BILL AND THANK THE CUSTOMER FOR VISITING
- END

PSEUDO CODE:

START

Input order

Initialize:

    order_item1,order_item2,order_item3,order_item4

Initialize Bill

Bill = order_item1 + order_item2 + order_item3 + order_item4

If customer_order is complete:

    Print Bill

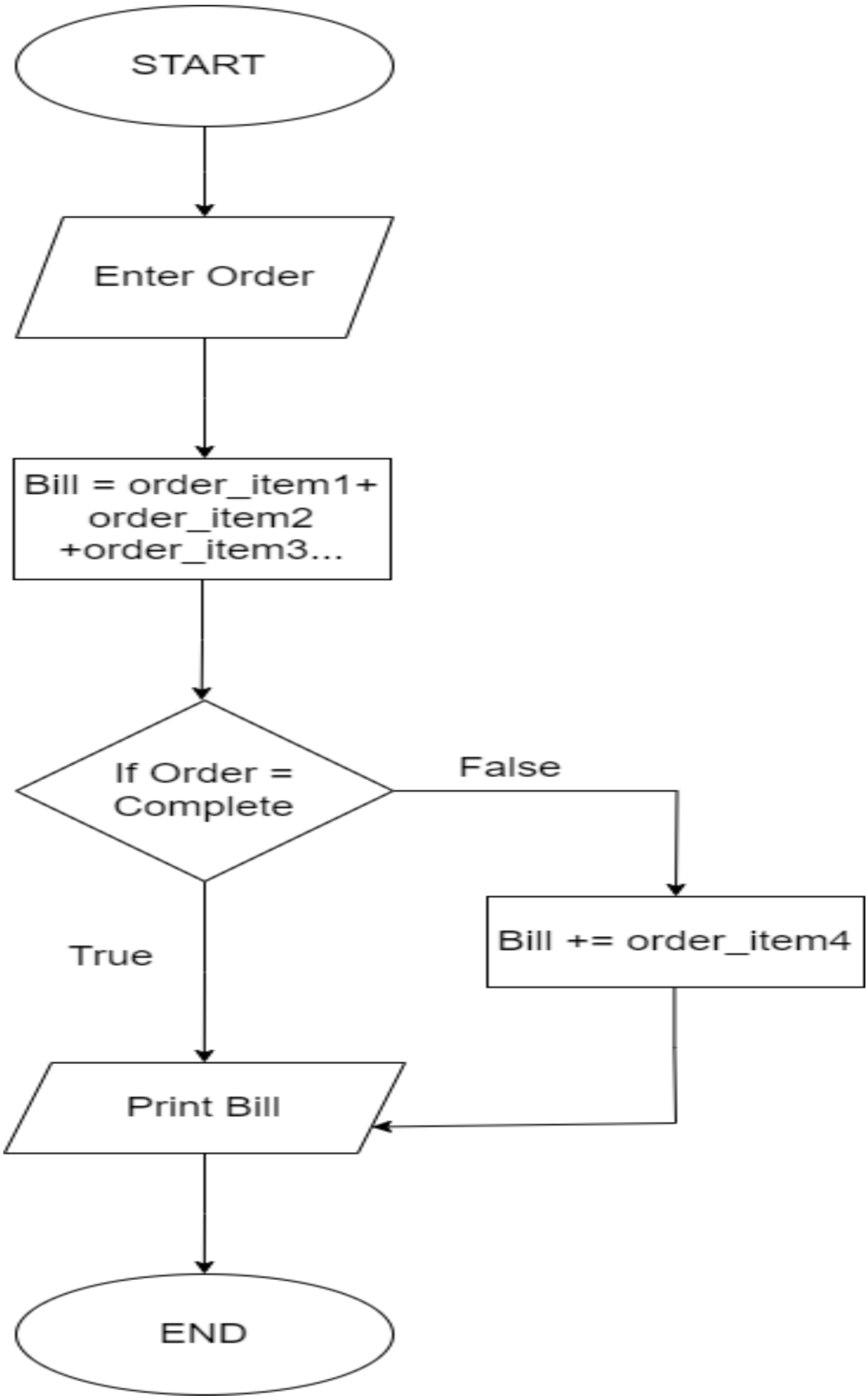
Else:

    Bill += order_item_5

    Print Bill

END

FLOW CHART:


START
Enter Order
Bill = order_item1+
order_item2
+order_item3...
If Order =
Complete
False
True
Bill += order_item4
Print Bill
END

TASK 2: ATM

ALGORITHM:

1. START
2. Input Amount for Deposit
3. IF Deposited Amount is less than 1000: Display “Amount is too low”
4. ELSE: Add the deposited amount to the current balance
5. Check account validity
6. IF account is valid: Display current balance
7. ELSE: Display “Account invalid”
8. END

PSEUDO CODE:

1. START
2. INT DEP_AMOUNT
3. INT ACCOUNT_VALIDITY
4. ACCOUNT_VALIDITY = TRUE
5. INT CURRENT_BALANCE
6. INPUT DEP_AMOUNT
7. IF DEP_AMOUNT <= 1000:
8. PRINT “AMOUNT IS TOO LOW “
9. ELSE: CURRENT_BALANCE += DEP_AMOUNT
10. IF ACCOUNT_VALIDITY == TRUE:
11. PRINT CURRENT_BALANCE
12. ELSE:
13. PRINT “ACCOUNT IS INVALID”
14. END

TASK 3: WHICH NUMBER IS GREATEST

ALGORITHM:

0. START
1. INPUT NUM1
2. INPUT NUM2
3. INPUT NUM3
4. INT GREATER_NUMBER
5. INT GREATEST_NUMBER
6. COMPARE NUM1 AND NUM2 AND ASSIGN GREATER TO GREATER_NUMBER
7. COMPARE GREATER_NUMBER TO NUM3 AND ASSIGN TO GREATEST_NUMBER
8. PRINT GREATEST_NUMBER

PSEUDO CODE:

1. Int num1,num2,num3,greater_num,
    a. greatest_num
2. Input num1,num2,num3
3. If num1>num2: greater_num = num1
4. Else: greater_num = num2
5. If num2>num3: greatest_num = num2
6. Else : greatest_num = num3
7. Print greatest_num

TASK 4:

ALGORITHM:

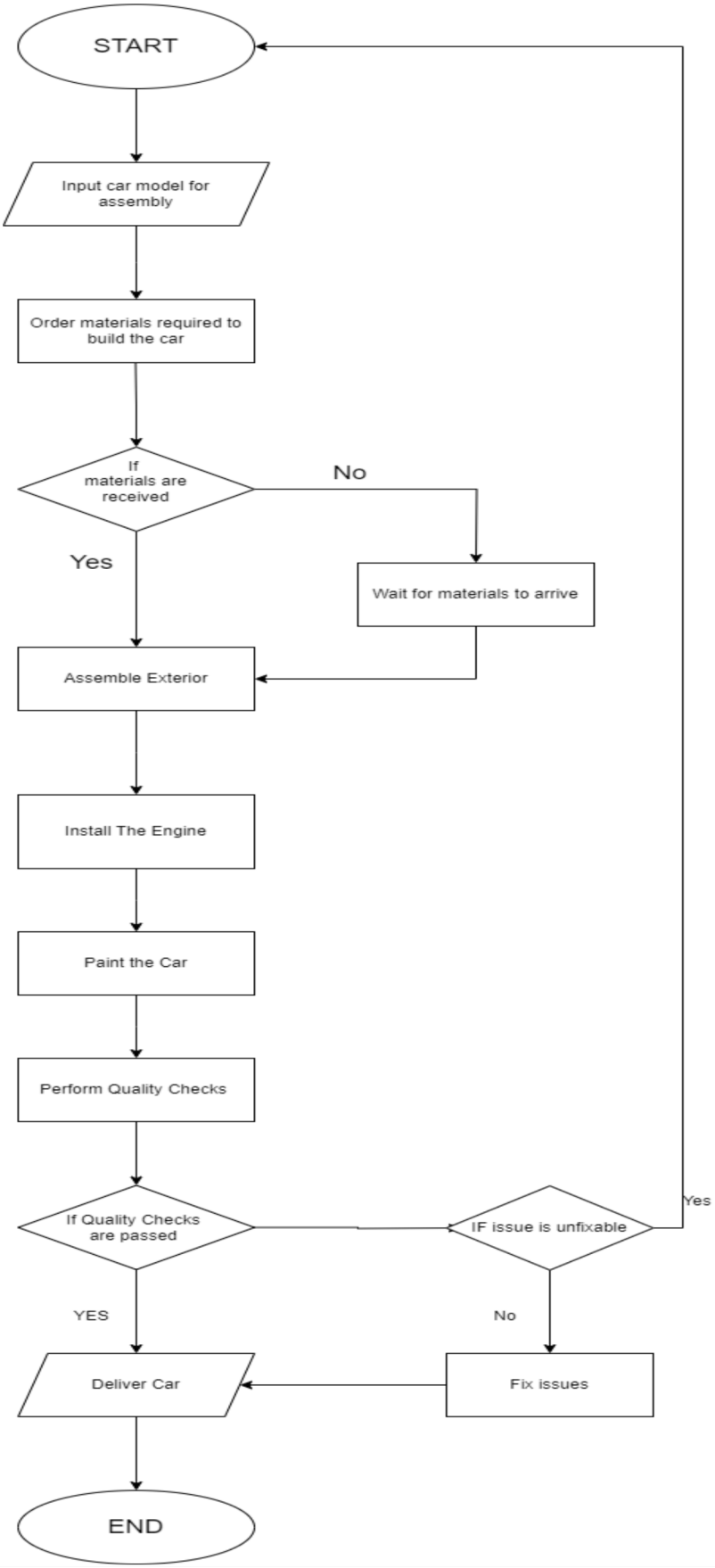
0. INPUT NUMBER BETWEEN 1 AND 12
1. 1=JANUARY,2=FEBRUARY,
2. 3=MARCH,4=APRIL,5=MAY,6=JUNE,
3. 7=JULY,8=AUGUST,9=SEPTEMBER,
4. 10=OCTOBER,11=NOVEMBER,
5. 12=DECEMBER
6. PRINT VALUE OF NUMBER

TASK 5: A SMALL CALCULATOR

PSEUDO CODE:

1. INT NUM1
2. INT NUM2
3. INT OPERATOR
4. INT RESULT
5. INPUT NUM1
6. INPUT NUM2
7. INPUT OPERATOR
8. IF OPERATOR == ‘+’:
9. RESULT = NUM1 + NUM2
10. PRINT RESULT
11. ELSE IF OPERATOR == ‘-‘:
    a. RESULT = NUM1 – NUM2
    b. PRINT RESULT
12. ELSE:
    a. PRINT”INVALID OPERATOR”

Task 06: Assemble A Car



TASK 07: Calculator

Algorithm:

- START
- Enter First Number
- Enter Second Number
- Choose an Operator between + , - , * , / , %
- If Operator is + :
    - Add First and Second Number and assign to Third Number
    - Print Third Number
- Else If Operator is - :
    - Subtract First Number From Second Number and assign to Third Number
    - Print Third Number

- If Operator is * :
    - Multiply First and Second Number and assign to Third Number
    - Print Third Number
- Else If Operator is / :
    - Divide First Number From Second Number and assign to Third Number
    - Print Third Number
- Else If Operator is % :
    - Take Modulus of First and Second Number and assign to Third Number
    - Print Third Number
- END

TASK 09: Why do we use .gitignore

We use .gitignore to make sure that some files are not committed to our github repo in order to prevent sensitive and confidential information like passwords from leaking to unwanted users. It is also useful to avoid committing unnecessary bloatware to our github repository

---

TASK 10: Difference between Pseudo Code and Algorithm:

Algorithm	Pseudo Code
Algorithm is a sequence of a solution	It is a representation of the Algorithm
There are no rules to writing Algorithms	Certain Programming Rules and Fundamentals need to be followed while writing Pseudo Codes
Algorithms are more specific	Pseudo Codes are more general
It is a plan for solving a problem	It provides an overview for the solution's logic.