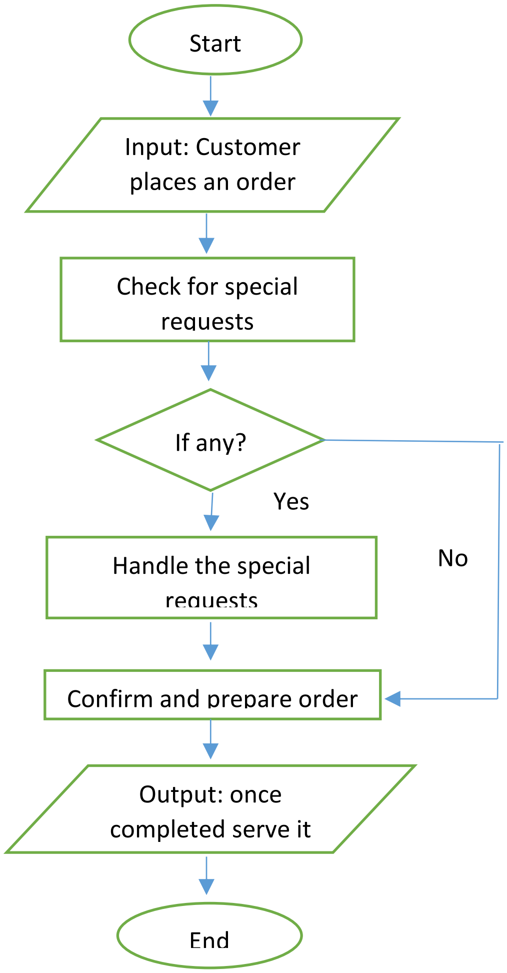
LAB 02 PF TASK

1. Design a flowchart, pseudocode, and algorithm for processing a customer order at a restaurant, including handling special requests (like add-ons).

FLOWCHART


Start
Input: Customer places an order
Check for special requests
If any?
Yes
No
Handle the special requests
Confirm and prepare order
Output: once completed serve it
End

PSEUDOCODE

```
START

// Input/Output
INPUT order

// Process
CHECK for special request

// Conditional Statements
IF AnySpecialRequest
    PRINT “handle request”
```

```
ELSE
   PRINT “confirm and prepare order”

OUTPUT serve
END
```

ALGORITHM

1) Start.
2) Ask the customer to place an order.
3) Check for special requests.
4) If any then handle them.
5) Otherwise confirm and prepare the order.
6) Once completed serve it.
7) End.

2. Design a pseudocode and algorithm for handling a customer’s deposit transaction at a bank, including checks for account validity and deposit amount conditions.

PSEUDOCODE

```
START

// Input/Output
INPUT Customer Account Information
INPUT Deposit Amount

// Conditional Statements
IF IsAccountValid then
IF DepositAmountValid then

// Process
SET the deposit
UPDATE Account balance

PRINT “Successful Deposit.”

END
```

ALGORITHM

1) Start.
2) Ask the user to input their account ID and deposit amount.
3) Using account ID, access the account details.
4) Check the validity of the account and deposit amount.
5) Update account balance.
6) Successful deposit.
7) End.

3. Design a pseudocode and algorithm to determine which of the three provided numbers is the greatest.

PSEUDOCODE

```
START

// Input/Output
INPUT n1
INPUT n2
INPUT n3

// variables and initialisation
SET greatest = n2

// Conditional Statements
IF n1 > greatest then
  SET greatest = n1

ELSE
IF n3 > greatest the
   SET greatest = n3

OUTPUT “the greatest number is:  “.

END
```

ALGORITHM

1) Start
2) Take three numbers (n1, n2, n3) as input.
3) Compare n1 and n2.
4) If n1 is greater than n2, go to step 6.
5) But if n2 is greater than n1, go to step 9.
6) Compare n1 and n3.
7) If n1 is greater than n3, then n1 is greatest number.
8) But if n3 is greater than n1, so n3 is the greatest number.
9) Compare n2 and n3.
10) If n2 is greater than n3, then n2 is the greatest number.
11) But if n3 is greater than n1, so n3 is the greatest number.
12) End.

4. Implement an algorithm where the user enters a number and displays an appropriate month.

ALGORITHM

1) Start.
2) Initialize a list of numbers from 1 to 12 with the corresponding months.
3) Ask the user to enter a number from 1 to 12.
4) Check if the number is in range of 1 to 12.
5) If the number is valid, proceed to step 7.
6) If the entered number is not within the range then, go back to step 3.
7) Retrieve the month name according to the number.
8) Display the month name to the user.
9) End.

5. Create a pseudocode of a small calculator which only does ‘+’ or ‘-‘operations.

PSEUDOCODE

```
1)  Start.
2)  Enter the first number.
3)  Read num1.
4)  Enter the second number.
5)  Read num2.
6)  Enter the operator (‘+’ or ‘-‘).
7)  Read op.
8)  If (op == ”+”)
    Print num1+num2 is equal to the sum.
9)  Else If (op == ”-”)
    Print num1-num2 is equal to the difference.
10) Else
    Print invalid.
11) End.
```

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

ALGORITHM

1) Start.
2) Prompt the user to enter two numbers (n1 and n2).
3) Ask for the operator (op) to be used which can be either of the five options (+, -, *, /, %).
4) Check the operator (op).
5) If op is ‘+’, then add n1 and n2.
6) Else if op is ‘-‘, then subtract n1 and n2.
7) Else if op is ‘*‘, then multiply n1 and n2.
8) Else if op is ‘/‘, then divide n1 and n2.
9) Else if op is ‘%‘, then perform integer division of n1 and n2.
10) Else, display an error message (invalid operator).
11) End.

7. Why do we use .gitignore?

A .gitignore file in a Git repository specifies which files and folders Git should ignore. It keeps the repository clean by disregarding files that should not be included in version control, such as temporary files. To avoid being tracked, files containing private information can be placed to the .gitignore file.

8. Difference between algorithm and pseudocode?

Algorithms can sometimes get hard to understand while pseudocode are easy to interpret. The process of debugging is simpler in pseudocode as compared to in algorithm. An algorithm is a step by step procedure to solve a given problem while a pseudocode is a method of writing an algorithm.