

TASK 2

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

PSEUDOCODE:

Step1: Start

Step2: Display "welcome to KFC"

Step3: Display "menu"

Step4: Read "add on request"

Step5: Read "Take order"

Step6: Calculate bill

Step7: Display "amount of bill"

Step8: Read "bill"

Step9: Display "please wait your order is preparing"

Step10: Display "order"

Step11: Display "Thank you for visiting KFC"

Step12: END.

Algorithm:

- Say customer to welcome to KFC
- Show menu to customer
- Take order from customer
- Add on request
- Calculate bill of customer
- Get bill from customer
- Say to customer wait few minutes your order is preparing
- Give his or her order
- Say thank you for visiting KFC
- END.

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:

Step1: Start

Step2: Display "Bank account"

step3: Display "send money"

step4: Read "send money"

step5: Display "easy paisa transfer"

step6: Display "bank transfer"

step7: Display "jazz cash transfer"

step8: Read "Bank transfer"

step9: Display "Bank-alHabib"

step10: Display "Bank allied"

step11: Read "Bank allied"

step12: Display "enter the IBAN number"

step13: Read "IBAN number"

step14: If

IBAN number is correct then

Step15: Display “enter the amount”

Step16: Read “amount”

Step17: If

Amount \leq available account balance in bank account then
Display “amount deposited successfully”

Step18: else if

Display “insufficient balance for this transaction”

Step19: ELSE

Display “you entered IBAN is incorrect”

Step20: End.

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

PSEUDO CODE:

Step1: START

Step2: Declare n1, n2, n3;

Step3: Display "Enter three numbers"

Step4: Read "n1&n2,&n3"

Step5: If

- $n1 > n2$ and $n1 > n3$, then
- Display "n1 is greatest"

Step6: Else If

- $n2 > n1$ and $n2 > n3$, then
- Display "n2 is greatest"

Step7: Else

8. Display "n3 is greatest"

Step9: End.

ALGORITHM:

- Start
- Declare n_1, n_2, n_3
- Output Enter three numbers
- Input three numbers for n_1, n_2, n_3
- If
 - $n_1 > n_2$ and $n_1 > n_3$ then
 - n_1 is greatest
 - else if
 - $n_2 > n_1$ and $n_2 > n_3$
 - n_2 is greatest
 - else
 - n_3 is greatest
- Output greatest number
- End.

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

Algorithm:

Step1: Start

Step2: print enter the number

Step3: read n

Step4: if

n==1 then

print January

Step5: else if

n==2 then

print February

step6: else if

n==3 then

print March

step7: else if

n==4 then

print April

step8: else if

n==5 then

print May

step9: else if

n==6 then

print June

step10: else if

n==7 then

print July

step11: else if

n==8 then

print August

step12: else if

n==9 then

print September

step13: else if

n==10 then

print October

step14: else if

n==11 then

print November

step15: else if

n==12 then

print December

step16: else

print N.O.T

step17: End.

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

Pseudo code:

Step1: start

Step2: read n1,n2,op, re,

Step3: print "enter the number for n1 and n2"

Step4: read "n1&&n2"

Step5: print "enter the operator, '+'&& '-' "

Step6: read "op"

Step7: if

Op == '+'

re=n1+n2

step8: else if

op== '-'

re=n1-n2

step9: else

print "N.O.T"

step10: End.

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

- Start
- Print "Enter the number for n1&&n2"
- Read "n1,n2";
- Print "Enter the operator, '+' && '-' && '*' && '/' && '%';
- Read "op"

If

- $op == '+'$
- $re = n1 + n2$

Else if

- $op == '-'$
- $re = n1 - n2$

Else if

- $op == '*'$
- $re = n1 * n2$

Else if

- $op == '/'$
- $re = n1 / n2$

Else if

- $op == '%'$
- $re = n1 \% n2$

Else

- Print “invalid”
 - End

7. Why we use gitignore ?

Ans: The purpose of gitignore files is to ensure that certain files not tracked by Git remain untracked . We use it to keep out things like:

- Temporary files that change often.
- Large or generated files that don't need to be in version control.
- Sensitive data like passwords.

This helps keep our Git repository clean and focused on the important code and files we want to share with others.

8. Difference between Algorithm and Pseudocode?

Pseudocode: Pseudocode is a simplified version of programming codes, written in plain English language and used to outline a program before its implementation.

Representation: Pseudocode uses a mix of natural language and programming constructs (such as loops and conditionals) to describe the algorithm. It is not executable but provides a clear outline of how the algorithm should be implemented.

Algorithm: An algorithm is a systematic, logical approach that provides a step-by-step procedure for computers to solve a specific problem.

- **Purpose:** To provide a clear and unambiguous set of instructions for solving a problem or completing a task.
- **Representation:** Algorithms are often represented in natural language, flowcharts, or mathematical notation. They do not require specific syntax or formatting.
- **Example:** An algorithm to sort a list of numbers might be described as follows:
 1. Compare the first two numbers.
 2. Swap them if the first is larger than the second.
 3. Move to the next pair of numbers and repeat step 2.
 4. Continue until the entire list is sorted.

