Lab 02

1. You are working in an e-commerce company and need to design a flowchart for processing an online order. The flowchart should include process modules for each step involved in handling an order and decision structures to handle stock availability and payment verification.

**Solution:**

**Steps:**

1. Read product name.
2. Check availability.
3. If product is available get payment else print “Not in Stock”
4. Verify Payment.
5. Print “order Placed”.

**Flowchart:**

Start

If Product is Available

Verify payment

Read

End

Start

Print “Not Available”

F

T

**Modules:**

Exit

Exit

Print “Payment Successful”

Verify payment

Get payment

Print “Cannot get Payment”

If payment is paid and non -zero

Read Product name

Read

F

T

**Pseudocode:**

1. Find if the number is multiple of 5.

**Code:**

1. START
2. // Input
3. INPUT Num1
5. // Process Steps
7. // Conditional Statements
8. IF **Num1 % 5 = 0** THEN
9. PRINT &*quot*; The number is multiple of 5 &*quot*;
10. ELSE
11. PRINT &*quot*; The number is not a multiple of 5 &quot;
12. END
13. Check if a character is uppercase or lowercase.

**Code:**

1. START
3. //Input
4. INPUT Char
6. //Process Steps
8. //Conditional Statement
9. IF **Char > A** AND **Char < Z** THEN
10. PRINT &*quot*; The Character is uppercase &*quot*;
11. ELSE
12. IF **Char > a** AND **Char < z** THEN
13. PRINT &*quot*; The Character is lowercase &*quot*;
14. ELSE
15. PRINT &*quot*; The Character is not an alphabet &*quot*;
16. END
17. Create a small calculator which only does ‘+’ or ‘\*‘Operations. (Hint: Take three variable inputs with one being used for the operator)

**Code:**

1. START
3. // Input
4. INPUT Num1
5. INPUT Operator
6. INPUT Num2
8. // variables and Initialization
9. SET Result to 0
11. // Process Steps
13. //Conditional Statements
14. IF **Operator =** “**+**” THEN
15. Result = Num1 + Num2
17. ELSE IF **Operator =** “**\***” THEN
18. Result = Num1 \* Num2
20. //Output
21. PRINT Result
23. END
24. Check whether a given number is positive, negative, or zero.

**Code:**

1. START
2. //Input
3. INPUT Num1
5. //Process Steps
7. //Conditional Statement
8. IF **Num1 > 0** THEN
9. PRINT &*quot*; The Number is positive &*quot*;
10. ELSE IF **Num1 = 0** THEN
11. PRINT &*quot*; The Number is zero &*quot*;
12. ELSE IF **Num1 < 0** THEN
13. PRINT &*quot*; The Number is negative &*quot*;
14. END
15. Figure out if a person is a teenager (between 13 and 19 years old).

**Code:**

1. START
2. //Input
3. INPUT Age
5. //Process Steps
7. //Conditional Statement
8. IF **Age > 13** AND **Age <19** THEN
9. PRINT &*quot*; The Person is teenager &*quot*;
10. END

**Algorithm:**

1. **Implement an algorithm to figure out if a given year is a leap year. A leap year is divisible**

**by 4, but not divisible by 100, except if it is also divisible by 400.**

**Solution:**

1. Start by checking if the year can be divided evenly by 4:
   1. If it cannot be divided evenly by 4, it's not a leap year.
   2. If it can be divided evenly by 4, continue to the next step.
2. Next, check if the year can be divided evenly by 100:
   1. If it cannot be divided evenly by 100, it is a leap year.
   2. If it can be divided evenly by 100, continue to the next step.
3. Finally, check if the year can be divided evenly by 400:
   1. If it can be divided evenly by 400, it is a leap year.
   2. If it cannot be divided evenly by 400, it's not a leap year.
4. **Implement an algorithm to count the number of occurrences of each character in a given**

**string.**

**Solution:**

1. Initialize an empty object named **Count** to store the counts of each character.
2. Start a loop to iterate through each character in the string.
   1. Check if the character already exists as a key in **Count** object
      1. If it exists, increment the value (count) of that key by 1.
      2. If it does not exist, add the character as a key to Count with an initial value of 1.
3. After the loop ends, the **Count** object will contain each character as a key and its occurrence count as the value.
4. Return or display the **Count** object to show the number of occurrences for each character.
5. **Write an algorithm to calculate x raised to the power y (i.e., xy) without using built-in power functions.**

**Solution:**

1. Initialize a variable result to 1. This variable will hold the result of **xy**.
2. If **y** is 0, return result because any number raised to the power of 0 is 1.
3. If **y** is positive, repeat the following steps **y** times:
   1. Multiply result by **x** and store the result back into result.
4. If **y** is negative, repeat the following steps ∣**y**∣ *(absolute value of Y)* times:
   1. Multiply result by **x** and store the result back into result.
   2. After the loop, set result to 1 / result since **x-y** is equivalent to .
5. Display Result
6. **Calculate the area of a circle given its radius r.**

**Solution:**

1. Take input from the user for the radius **r** of the circle.
2. Set the value of **π** to 3.14159.
3. Calculate the area **A** of the circle using the formula: **A= π × r × r**
4. Display the calculated area **A** to the user.
5. **Find the median of three given numbers.**

**Solution:**

* + - 1. Take three numbers **A**, **B**, and **C** as input.
      2. Compare the three inputs:
         1. If (A ≤ B ≤ C) or (C ≤ B ≤ A) then **B** is median.
         2. Else If (B ≤ A ≤ C) or (C ≤ A ≤ B) then **A** is median.
         3. Else **C** is median.
      3. Display median value.

**Github:**