**BY:(Muhammad Ibrahim Khawer)**

**Task-1**

DECLARE alphabets, str1 As String

DECLARE i : Integer

DECLARE isPangram : Boolean

DECLARE nextChar : Char

alphabets ← "abcdefghijlmnopqrstuvwxyz"

i ← 0

isPangram ← True

nextChar ← "

OUTPUT"Enter string: "

INPUT str1

str1 ← LOWER(str1)

FOR i ← 1 To LEN(alphabets)

nextChar ← MID(alphabets, i, 1)

IF LOCATE(str1, nextChar) ← 0 THEN

isPangram ← False

END IF

NEXT i

IF isPangram ← True Then

OUTPUT"Input string is pangram."

ELSE

OUTPUT"Input string is NOT a pangram."

END IF

**Task-2**

DECLARE Str1, Str2, Str3, Character, Word : String

DECLARE Count : Integer

Str1 ← ""

Str2 ← ""

Str3 ← ""

Character ← ""

Word ← ""

Count ← 0

OUTPUT "Enter a string: "

INPUT Str1

OUTPUT "Enter Character to Remove: "

INPUT Str2

OUTPUT "Enter Character to Add: "

INPUT Str3

FOR Count ← 1 To LEN(Str1)

Character ← MID(Str1, Count, 1)

IF Character <> Str2 Then

Word ← Word + Character

ELSE

Character ← Str3

Word ← Word + Character

END IF

NEXT Count

OUTPUT "Updated String: " + Word

**Task-3**

DECLARE Str1 : String

DECLARE Char1 : Char

DECLARE i, alphaCount, digitCount : Integer

'INITIALISATION

Str1 ← ""

Char1 ← "

i ← 0

alphaCount ← 0

digitCount ← 0

'INPUT

OUTPUT "Enter string to process: "

INPUT Str1

'PROCESS

FOR i ← 1 To LEN(Str1)

Char1 ← MID(Str1, i, 1)

IF Char1 >= "A" AND Char1 <= "Z" THEN

alphaCount ← alphaCount + 1

ELSEIF Char1 >= "a" AND Char1 <= "z" THEN

alphaCount ← alphaCount + 1

ELSEIF Char1 >= "0" AND Char1 <= "9" THEN

digitCount ← digitCount + 1

END IF

NEXT

'OUTPUT

OUTPUT "Alphabets: " + alphaCount

OUTPUT "Digits: " + digitCount

**Task-4**

DECLARE str1, str2, str3, GreatestChar, space : String

DECLARE Char1, Char2 : Char

DECLARE i, count : Integer

str1 ← ""

str2 ← ""

str3 ← ""

space ← " "

Char1 ← "

Char2 ← "

GreatestChar ← "

i ← 0

count ← 0

OUTPUT "Enter string: "

INPUT str1

str1 = LOWER(str1)

FOR i ← 1 To LEN(str1)

Char1 ← MID(str1, i, 1)

FOR count ← 1 To LEN(str1)

Char2 ← MID(str1, count, 1)

IF Char2 ← Char1 THEN

str2 ← str2 + Char1

END IF

NEXT

IF LOCATE(GreatestChar, Char1) ← 0 THEN

IF LEN(str3) < LEN(str2) THEN

GreatestChar ← Char1

END IF

IF LEN(str2) > LEN(str3) THEN

str3 ← str2

END IF

END IF

str2 = ""

NEXT

OUTPUT" Most reoccurring character is " + GreatestChar

**Task-5**

'DECLARATION

DECLARE Str1, Str2, Str3 : String

DECLARE i, count, num1 : Integer

DECLARE Char1, Char2 : Char

'INITIALISATION

Str1 ← ""

Str2 ← "AEIOU"

Str3 ← ""

i ← 0

num1 ← 0

count ← 0

Char1 ← "

Char2 ← "

'INPUT

OUTPUT "Enter string to process: "

INPUT Str1

Str1 ← UPPER(Str1)

'PROCESS

FOR i ← 1 To LEN(Str1)

Char1 ← MID(Str1, i, 1)

FOR count ← 1 To LEN(Str1)

Char2 ← MID(Str1, count, 1)

IF Char2 ← Char1 THEN

num1 ← num1 + 1

END IF

NEXT

IF LOCATE(Str3, Char1) ← 0 THEN

Str3 ← Str3 + Char1

IF LOCATE(Str2, Char1) THEN

OUTPUT Char1 + " = " + num1)

END IF

END IF

num1 ← 0

NEXT

**Task-6**

DECLARE count : Integer

count ← 0

OUTPUT"Enter The Amount Of Donouts: "

INPUT count

IF count < 10 Then

OUTPUT "Number Of Donouts: " & count

Else

OUTPUT "Number Of Donouts: many"

END IF

**Task-7**

DECLARE str1, str2, str3 : String

str1 ← ""

str2 ← ""

str3 ← ""

OUTPUT "Enter String To Process: "

INPUT str1

IF LEN(str1) > 2 THEN

str2 ← LEFT(str1, 2)

str3 ← RIGHT(str1, 2)

OUTPUT "Processed String: " + str2 + str3)

ELSE

OUTPUT "Processed String: " & str1

END IF

**Task-8**

'DECLARATION

DECLARE Str1, Str2, Str3 : String

DECLARE Count, Letters : Integer

DECLARE Char1, Char2, Char3 : Char

'Initialisation

Str1 ← ""

Str2 ← ""

Str3 ← "\*"

Char1 ← "

Char2 ← "

Char3 ← "

Count ← 0

Letters ← 0

'INPUT

OUTPUT "Enter The String To Process: "

INPUT Str1

Str1 ← LOWER(Str1)

'PROCESS

Char1 ← LEFT(Str1, 1)

FOR Count ← 1 To LEN(Str1)

Char2 ← MID(Str1, Count, 1)

IF Char2 ← Char1 THEN

Char2 ← Str3

END IF

Letters ← Letters + 1

IF Letters > 1 THEN

Str2 ← Str2 + Char2

END IF

NEXT

Char3 ← Char1

OUTPUT "Processed String: " + Char3 + Str2

**Task-9**

DECLARE Str1, Str2, Str3, Str4, Str5, Str6 : String

'Initialisation

Str1 ← ""

Str2 ← ""

Str3 ← ""

Str4 ← ""

Str5 ← ""

Str6 ← ""

'INPUT

OUTPUT "Enter First String: "

INPUT Str1

OUTPUT "Enter Scond String: "

INPUT Str2

'PROCESS

Str3 ← LEFT(Str1, 2)

Str4 ← LEFT(Str2, 2)

Str5 ← MID(Str1, 3, LEN(Str1))

Str6 ← MID(Str2, 3, LEN(Str2))

OUTPUT "First String: " + Str4 + Str5)

OUTPUT "Second String: " + Str3 + Str6)

**Task-10**

DECLARE Str1, Str2, Str3, Str4 : String

Str1 ← ""

Str2 ← "ing"

Str3 ← "ly"

Str4 ← ""

OUTPUT "Enter String: "

INPUT Str1

Str4 ← RIGHT(Str1, 3)

IF LEN(Str1) >= 3 AND Str4 <> Str2 THEN

OUTPUT "Updated String: " + Str1 + Str2

ELSEIF Str4 = Str2 Then

OUTPUT "Updated String: " + Str1 + Str3

ELSE

OUTPUT "String Was Very Short: " + Str1

END IF

**Task-11**

DECLARE str1, str2 : String

DECLARE num1, num2 : String

DECLARE char1 : Char

str1 ← ""

str2 ← ""

char1 ← "

num1 ← 0

num2 ← 0

OUTPUT "Enter string: "

INPUT str1

num1 ← LOCATE(str1, "not")

num2 ← LOCATE(str1, "bad")

IF LOCATE(str1, "not") > 0 AND LOCATE(str1, "bad") > 0 THEN

IF num2 > num1 THEN

num2 ← num2 + 3

num1 ← num1 - 1

FOR i ← 1 To LEN(str1)

char1 ← MID(str1, i, 1)

IF i > num1 And i < num2 THEN

IF LOCATE(str2, "good") = 0 THEN

str2 ← str2 + "good"

END IF

ELSE

str2 = str2 + char1

END IF

NEXT

END IF

END IF

OUTPUT "Updated String: " + str2