Computer programming assignments

**Find if the input string has all the alphabets; capital or small alike.**

Str1, Str2 As String

ThisChar As Char

SpaPos As Integer

Str1 = ""

Str2 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

ThisChar = ‘’

INPUT("Enter string to be processed: "),Str1

Str1 = UCase(Str1)

For a = 1 To Len(Str1)

ThisChar = Mid(Str1, a, 1)

SpaPos = InStr(Str2, ThisChar)

If SpaPos = 0 Then

PRINT("The string dosen't have all alphabets")

End

End If

Next

PRINT("Entered string has all alphabets")

**Replace selected character with another in entered string.**

Str1 🡪 ""

Str2 🡪 ""

Char1 🡪 ""

Char2🡪 ""

nextChar 🡪""

Count🡪 0

INPUT("Enter String: ") ,Str1

("Enter character to remove: "), Char1

("Enter character to replace: ") , Char 2

For Count --> 1 To Len(Str1)

nextChar --> Mid(Str1, Count, 1)

IF

nextChar --> Char1

THEN

nextChar --> char2

End If

Str2 --> Str2 + nextChar

Next

PRINT ("Final string --> " + Str2)

**Count and output the NUMBER of an entered character in a string. Also output separate counts for alphabets (cap & small together), digits and other characters in same entered string.**

AlphaStr --> "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

DigitStr --> "0123456789"

aCount --> 0

dCount --> 0

INPUT("Input string to be processed: "),Str1

For count --> 1 To Lenght(str1)

thisChar --> Mid(str1, count, 1)

If InStr(AlphaStr, UCase(thisChar)) Then

aCount --> aCount + 1

Else

If InStr(DigitStr, thisChar) Then

dCount --> dCount + 1

End If

Next

oCount --> Len(str1) - aCount - dCount

PRINT("Number of characters in string : " + Len(str1))

PRINT("Number of alphabets in string :" + aCount)

PRINT("Number of digits in string :" + dCount)

PRINT("Number of other characters in string :" + oCount)

**Find the character that appears most number of times in an entered string and output it.**

myChar, thisChar, nextChar As Char

Highest, Count, a, b As Integer

Str1 As String

myChar = ‘’

Highest = 0

Count = 0

INPUT("EnterString to be processed: "),Str1

For a = 1 To Len(Str1)

nextChar = Mid(Str1, a, 1)

Count = 0

For b = 1 To Len(Str1)

thisChar = Mid(Str1, b, 1)

If thisChar = nextChar Then Count = Count + 1

Next

If Count > Highest Then

Highest = Count

myChar = nextChar

End If

Next

PRINT(myChar + " appeared " + Highest +" times.")

**Find the count of vowels characters in an entered string separately.**

VowStr = "AEIOUaeiou"

VowTimes = 0

INPUT("Enter string to be processed: "),Str1

For a = 1 To Lenght(Str1)

ThisChar = Mid(Str1, a, 1)

If InStr(VowStr, ThisChar) Then

VowTimes = VowTimes + 1

End If

Next

PRINT(" Number of vowels in entered string are: " + VowTimes)

**donuts:**

**INPUT an INT count of a number of donuts, OUTPUT a string of the form 'Number of donuts: <count>', where <count> is the number input. However, if the count is 10 or more, then use the word 'many' instead of the actual count. So donuts(5) outputs 'Number of donuts: 5' and donuts(23) outputs 'Number of donuts: many'**

Count As Integer

INPUT ("Enter number of donuts: "), Count

If Count >= 10 Then

PRINT("Number of donuts:many ")

Else

PRINT("Number of donuts: " & Count)

End If

**both\_ends: Input a string s, output a string made of the first 2 and the last 2 chars of the original string, so 'spring' yields 'spng'. However, if the string length is less than 2, return string s instead the empty string.**

Str1, FirstChars, LastChars As String

INPUT(" Enter string to be processed: "),Str1

If Len(Str1) <= 2 Then

PRINT(Str1)

End

End If

FirstChars = Left(Str1, 2)

LastChars = Right(Str1, 2)

PRINT("Processed string is: " + FirstChars + LastChars)

**Fix Start:**

**Given a string s, return a string where all occurences of its first char have been changed to '\*', except do not change the first char itself. e.g., 'babble' yields 'ba\*le' or 'alpha' outputs 'alph'. Assume that the string is length 1 or more.**

Str1 As String

FirstChar As Char

ThisChar As Char

INPUT("Enter string to be processed: "),Str1

FirstChar = Left(Str1, 1)

PRINT(FirstChar)

For COUNT = 1 To Len(Str1) - 1

ThisChar = Mid(Str1, (COUNT + 1), 1)

If ThisChar = FirstChar Then

ThisChar = "\*"

End If

PRINT(ThisChar)

Next

**Mix Up:**

**Given strings a and b, return a single string with a and b separated by a space '<a> <b>', except swap the first 2 chars of each string. e.g 'mix', pod' -> 'pox mid' 'dog', 'dinner' -> 'dig donner' Assume a and b are length 2 or more.**

Declare Str1, Str2, FS, SS, a, b As String

Declare SpaPos1, SpaPos2 As Integer

INPUT("Enter the first string: "),Str1

INPUT("Enter the second string: "),Str2

If Len(Str1) <= 2 And Len(Str2) <= 2 Then

PRINT("Incorrect lenght of input")

End

End If

FS = Left(Str1, 2)

SS = Left(Str2, 2)

SpaPos1 = InStr(Str1, " ")

SpaPos2 = InStr(Str2, " ")

a = Right(Str1, SpaPos1 - 2)

b = Right(Str2, SpaPos2 - 2)

Str1 = SS & a

Str2 = FS & b

PRINT(Str1 & Str2)

**verbing:**

**Given a string, if its length is at least 3, add 'ing' to its end. Unless it already ends in 'ing', in which case add 'ly' instead. If the string length is less than 3, leave it unchanged. Return the resulting string. examples:**

**input 'end' --> 'ending'**

**input 'ending' --> 'endingly'**

**input 'go' --> 'go'**

Str1, Str2 As String

INPUT("Enter sting to be processed: "), Str1

If Len(Str1) < 3 Then

PRINT(Str1)

End

End If

If Right(Str1, 3) = "ing" Then

Str2 = Str1 + "ly"

Else

Str2 = Str1 + "ing"

End If

PRINT("Processed string is: " & Str2)

**not\_bad:**

**Given a string, find the first appearance of thesubstring 'not' and 'bad'. If the 'bad' follows the 'not', replace the whole 'not'...'bad' substring with 'good'. Return the resulting string.**

**Input: 'This dinner is not that bad!'**

**Outputs: This dinner is good!**

Declare NotPos, BadPos As Integer

Declare FP, SP, str1 As String

INPUT("Enter line to be processed: "),str1

NotPos = InStr(str1, "not")

BadPos = InStr(str1, "bad")

FP = Left(str1, NotPos - 1)

SP = Right(str1, (Len(str1) - BadPos - 4))

PRINT(FP + " good " + SP)