Aula de 05-12-2023

Aparecem feitos os que estão a

\_\_\_

Fazer os que estão a

\_\_\_

Escrever cada letra de um texto? (Algoritmo + Python)

Por exemplo: “este texto?”

e

s

t

e

t

e

x

t

o

Algoritmo Letras

Este algoritmo, dado uma frase, calcula e escreve as suas letras.

L10 [Inicializar alfabeto]

Alfab🡨”abc...z”

L20 [Ler e validar frase]

DO

\_\_\_PRINT(“Qual a frase?”)

\_\_\_READ(Frase)

UNTIL Frase<>””

L30 [Estabelecer ciclo para processar carateres da frase]

DO L40 TO L60 FOR I=1 TO LENGHT(Frase)

\_\_ L40[Extrair caracter]

\_\_\_car🡨SUB(Frase,i,1)

\_\_\_L50 [Calcular a posição do carater no alfabeto]

\_\_\_Onde🡨INDEX(Alfab,car)

\_\_\_L60 [Escrever o carater se for letra]

\_\_\_IF Onde<>0

\_\_\_THEN Print (Car)

L70 [Terminar]

Exit []

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Alfab | Frase | I | car | Onde | Onde<>0 | Saida |
| L10 | “abc…z” |  |  |  |  |  |  |
| L20 |  | “este texto?” |  |  |  |  | Qual a frase? |
| L30 |  |  | 1 | “e” | 5 | 5<>0V | “e” |
|  |  |  | 2 | “s” | 19 | 19<>0V | “s” |
|  |  |  | … |  |  |  |  |
|  |  |  | 11 | “?” | 0 | 0<>0F |  |

#Este algoritmo, dado uma frase, calcula e escreve as suas letras.  
#L10 [Inicializar alfabeto]  
alfa="abcdefghijklmnopqrstuvwxyz"  
#L20 [Ler e validar frase]  
print("Qual o texto?")  
Texto=input()  
while len(Texto)==0:  
 print("Qual o texto?")  
 Texto = input()  
# L35 [Calcular tamanho do texto]  
tam=len(Texto)  
# L30 [Establecer ciclo para processar carateres da frase]  
for i in range(tam):  
 #L40[Extrair caracter]  
 letra = Texto[i]  
 #L50 [Calcular a posição do carater no alfabeto]  
 Onde=alfa.find(letra)  
 #L60 [Escrever o carater se for letra, atualizando numero de letras]  
 if Onde!=-1:  
 print("Carater nº", i + 1, ") = ", letra)  
#L70 [Terminar]

Exercício 1. Escrever o nº de vezes que cada letra aparece?

Por exemplo: “aula de hoje”

a ocorre 2 vezes

d ocorre 1 vezes

e ocorre 2 vezes

h ocorre 1 vezes

j ocorre 1 vezes

l ocorre 1 vezes

o ocorre 1 vezes

u ocorre 1 vezes

Algoritmo TodaFrqLetras

Este algoritmo dado um texto, escreve a frequência de cada uma das letras.

T10[Ler e validar texto]

DO

PRINT(“Texto?”)

READ(Texto)

UNTIL Texto<>””

F20 [Inicializar letras do alfabeto]

Alfabeto🡨‘"abcdefghijklmnopqrstuvwxyz"  
F30 [Determinar tamanho do texto]

TamanhoTexto🡨 LENGTH(Texto)

F40 [Inicializar vector de frequencias]

DO FOR i = 1 to 26

Frequencia [ i ] 🡨 0

F50 [Estabelecer ciclo para processar os caracteres]

DO F60 TO F80 FOR i = 1 to TamanhoTexto

F60 [Extrair caracter]

Caracter 🡨 SUB (Texto, I, 1)

F70 [Determinar posição do caracter no alfabeto]

onde 🡨 INDEX(Alfabeto, Caracter)

F80 [Actualizar vector de frequências caso seja letra]

IF onde<>0

THEN Frequencia[onde]🡨Frequencia[onde]+1

F90 [Estabelecer ciclo para escrever as frequências das letras]

DO F100 FOR i = 1 to 26

F100 [Escrever frequência de letra caso exista]

IF Frequencia[i]>0

THEN PRINT ( ‘A frequência letra ’, SUB(Alfabeto, i, 1), ‘ é = ‘, Frequencia [ i ] )

F110 [Terminar]

EXIT []

#Ler e validar um texto  
print("Qual o texto?")  
Texto=input()  
while len(Texto)==0:  
 print("Qual o texto?")  
 Texto = input()  
#Dado um texto contar o número de vezes que cada letra ocorre.  
#Inicializar vetor das frequencias  
L=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]  
#Inicializar as letras "possiveis"  
alfa="abcdefghijklmnopqrstuvwxyz"  
#Determinar o tamanho do taxto  
tam=len(Texto)  
#Estabelecer um ciclo para processar as letras  
for i in range(tam):  
 #Extrair cada letra  
 letra= Texto[i]  
 # Encontrar a letra nas letras "possiveis"  
 onde = alfa.find(letra)  
 # Atualizar vetor das frequencias caso a letra seja uma das "possiveis"  
 if onde >= 0:  
 L[onde] = L[onde] + 1  
# Estebelecer um ciclo para processar todas a letras do alfabeto  
for i in range(len(alfa)):  
 # Se essa letra existe no texto  
 if L[i] > 0:  
 print(alfa[i], "ocorre ", L[i], "vezes")

Outro Exercício: Dado um texto, escreva cada uma das suas as palavras.

Exemplo: “a arvore da ana e a arvore da alda”.

a

arvore

da

ana

e

a

arvore

da

alda

Algoritmo PalavrasTxt

Este algoritmo dado um texto escreve cada uma das suas palavras.

P10[Ler e validar texto]

DO

PRINT(“Texto?”)

READ(Texto)

UNTIL Texto<>””

P20 [Acrescentar um espaço no fim do texto]

Texto🡨Texto+” “

P30 [Calcular a posição do 1º espaço]

Esp🡨INDEX(Texto, “ “)

P40 [Estabelecer ciclo para processar palavras]

DO P50 TO P75 WHILE Esp<>0

P50 [Extrair palavra]

Pal🡨SUB(Texto,1, Esp-1)

P60 [Escrever palavra]

PRINT(Pal)

P70 [Actualizar o texto]

Texto🡨SUB(Texto, Esp+1)

P75 [Calcular a posição do 1º espaço]

Esp🡨INDEX(Texto, “ “)

P80 [Terminar]

Exit []

print("Qual o texto?")  
Texto=input()  
while len(Texto)==0:  
 print("Qual o texto?")  
 Texto = input()  
Txt=Texto+" "  
esp=Txt.find(" ")  
while esp>=0:  
 print(Txt[0:esp])  
 Txt= Txt[esp+1: len(Txt)]  
 esp = Txt.find(" ")

Exercício 2. Dado um texto, escreva a ocorrência de cada uma das suas as palavras.

Exemplo: “a arvore da ana e a arvore da alda”.

a 2

arvore 2

da 2

ana 1

e 1

alda 1

Algoritmo ContaFrequenciaPalavras\_Diferentes  
Este algoritmo, dado um texto, conta as ocorrências de cada uma das palavras.  
C10 [Ler e validar o texto]  
\_\_\_\_Do  
\_\_\_\_\_\_\_Print("Qual o texto?")  
\_\_\_\_\_\_\_Read(texto)  
\_\_\_\_texto<>"“

C20 [Actualizar texto]  
\_\_\_\_texto<--texto+" "  
C30 [Inicializar contador de palavras]  
\_\_\_\_cont<--0  
C40 [Determinar primeiro espaço]  
\_\_\_\_Esp<--Index(texto," ")  
C50 [Estabelecer ciclo para processar palavras]  
\_\_\_\_Do C55 to C80 while Esp<>0

\_\_\_\_\_\_\_\_\_C55 [Extrair a palavra]

\_\_\_\_\_\_\_\_\_Pal <-- SUB(Texto,1,Esp-1)

\_\_\_\_\_\_\_\_\_C60 [Actualizar ocorrências de palavras]

\_\_\_\_\_\_\_\_\_P[cont+1] <-- Pal

\_\_\_\_\_\_\_\_\_oc[cont+1] <-- 1

\_\_\_\_\_\_\_\_\_qt<-- 1

\_\_\_\_\_\_\_\_\_DO WHILE P[qt]<>Pal and qt<cont

\_\_\_\_\_\_\_\_\_\_\_\_\_qt <-- qt+1

\_\_\_\_\_\_\_\_\_IF qt<=cont

\_\_\_\_\_\_\_\_\_THEN oc[qt] <-- oc[qt]+1

\_\_\_\_\_\_\_\_\_ELSE cont <-- cont+ 1

\_\_\_\_\_\_\_\_\_C70 [Actualizar texto]  
\_\_\_\_\_\_\_\_\_texto<--Sub(texto,Esp+1)  
\_\_\_\_\_\_\_\_\_C80 [Procurar próximo espaço]  
\_\_\_\_\_\_\_\_\_Esp<--Index(texto," “)

C90 [Escrever ocorrências de cada palavra]

IF cont>0

THEN DO FOR K=1 to cont  
\_\_\_\_  Print(P[k],"aparece no texto",oc[k],"vezes")  
C100[Terminar]  
\_\_\_\_Exit[]

#Inicializar vetor das palavras diferentes  
Pw=["","","","","","","","","","","","","","","","","","",""]  
#Incializar vetor das frequencias das palaras diferentes  
F=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]  
#Ler e validar o texto  
print("Qual o texto?")  
Texto=input()  
while len(Texto)==0:  
 print("Qual o texto?")  
 Texto = input()  
#Incializar o novotexto com o texto e um espaço no fim  
NovoTxt=Texto+" "  
#Inializar o numero de palavras diferentes  
numPals = 0  
#Procurar o primeiro espaço do texto  
esp=NovoTxt.find(" ")  
#Enquanto houver espaços  
while esp>=0:  
 #Estrair palavrara  
 pal = NovoTxt[0:esp]  
 # Colocar apalvara no fim do vetor de palavras e ocorrencia a 1  
 Pw[numPals] = pal  
 F[numPals] = 1  
 # Ver se a palavra já existia  
 j = 0  
 while Pw[j] != pal:  
 j = j + 1  
 # Se a palavra nao existia, entao incrementar o numero de palavras diferentes  
 # Senao atualizar a frequencia dessa palavra  
 if j == numPals:  
 numPals = numPals + 1  
 else:  
 F[j] = F[j] + 1  
 #Atualizar texto  
 NovoTxt= NovoTxt[esp+1: len(NovoTxt)]  
 #Encontrar o primeiro espaço do novo texto  
 esp = NovoTxt.find(" ")  
 #Escrever o numero de palavras diferentes  
 print (Texto, "tem ", numPals, " palavras diferentes ")  
#Escrever cada palavra e respetiva ocorrencia  
for i in range(numPals):  
 print( Pw[i]," ocorre ", F[i], "vezes")

**Aula de hoje**

Exercício 3. Dado um texto sem sinais de pontuação:

* Escreva o número de palavras do texto.
* Escreva a frequência de palavras que começam e terminam por uma letra do

Alfabeto, assim como o número e quais dessas palavras são diferentes

* Escreva o número de palavras que começam com uma vogal, assim como quantas e quais delas são diferentes

Exemplo:

Entrada

as saias da ana e as saias da ana andreia são saias de seda

a

Saída

Há 14 palavras no texto.

Começadas por a e terminadas por a há 3 palavras

Começadas por a e terminadas por a há duas palavras diferentes (ana, andreia)

Há 5 palavras começadas por vogais

Há 3 palavras diferentes começadas por vogais (as, ana, andreia)

Algoritmo Ex3

Este algoritmo dado um texto sem sinais de pontuação e uma letra, escreve o número de palavras do texto, a frequência de

palavras que começam, assim como começam e terminam por essa letra e o número de palavras que começam com uma vogal.

E10 [Ler e validar texto]

DO

PRINT(“Texto?”)

READ(Texto)

UNTIL Texto <>””

E15 [Ler e validar letra]

DO

PRINT(“Letra?”)

READ(Letra)

UNTIL Letra <>””

E20 [Estabelecer vogais]

Vogais🡨”aeiou”

E30 [Inicializar número de vogais]

Qtv🡨0

E40 [Estabelecer letras do alfabeto]

Alfa🡨”abc…z”

E50 [Inicializar número de palavras]

Qt🡨0

E60 [Inicializar ocorrências início e fim com letra]

Oc🡨0

E70 [Procurar primeiro espaço]

ESP🡨INDEX(texto, “ “)

E80 [Estabelecer ciclo enquanto houver espaços]

DO E90 TO E140 WHILE Esp<>0

E90 [Atualizar o número de palavras]

Qt🡨Qt+1

E100 [Extrair palavra]

Pal🡨SUB(Texto,1, Esp-1)

E110 [Atualizar o número de palavras começadas por vogais]

IF LEN(Pal)>1

THEN IF index (Vogais, Sub(Pal, 1,1))>0

THEN Qtv🡨Qtv+1

E120 [Atualizar ocorrências de palavras a começar e acabar com a letra escolhida]

IF LEN(Pal)>1

THEN IF Sub(Pal, 1,1)=Sub(Pal, len(Pal),1) and Sub(Pal, 1,1)=Letra

THEN Oc🡨Oc+1

E130 [Atualizar texto]

Texto🡨Sub(texto,Esp+1)

E140 [Procurar próximo espaço]

Esp🡨INDEX(texto, “ “)

E150 [Escrever número de palavras no texto]

PRINT(“Há “, qt, “ palavras no texto”)

E160 [Escrever número de palavras começadas e terminadas por letra]

PRINT (“A começar e a acabar por:”, letra,” há “,Oc)

E170 [Escrever número de palavras começadas por vogais no texto]

PRINT(“Há “, qtv, “ palavras começadas por vogais no texto”)

E180 [Terminar]

EXIT []

VetorPal = ["", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""]  
VetorOc = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]  
NPDCT=0  
VetorPalCV = ["", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""]  
VetorOcCV = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]  
NPDCV=0  
print("Qual o texto?")  
texto=input()  
while len(texto)==0:  
 print("Qual o texto?")  
 texto = input()  
print("Qual a letra?")  
letra=input()  
while len(letra)==0:  
 print("Qual a letra?")  
 letra = input()  
#Inicializar o numero de palavras começadas e terminadas por letra  
PCT=0  
vogais="aeiou"  
PCV=0  
texto2=texto+" "  
#V40 [Inicializar contador de palavras]  
Cont=0  
#V50 [Determinar primeiro espaço]  
Esp=texto2.find(" ")  
#V60 [Estabelecer ciclo para processar palavras]  
while Esp!=-1:  
 Pal = texto2[0:Esp]  
 print(Pal)  
 if Pal[0]==Pal[len(Pal)-1] and Pal[0]==letra:  
 PCT=PCT+1  
 VetorPal[NPDCT] = Pal  
 VetorOc[NPDCT] = 1  
 j = 0  
 while VetorPal[j] != Pal:  
 j = j + 1  
 if j == NPDCT:  
 NPDCT = NPDCT + 1  
 else:  
 VetorOc[j] = VetorOc[j] + 1  
 if vogais.find(Pal[0])>=0:  
 PCV = PCV+1  
 VetorPalCV[NPDCV] = Pal  
 VetorOcCV[NPDCV] = 1  
 j = 0  
 while VetorPalCV[j] != Pal:  
 j = j + 1  
 if j == NPDCV:  
 NPDCV = NPDCV + 1  
 else:  
 VetorOcCV[j] = VetorOcCV[j] + 1  
 Cont=Cont+1  
 texto2 = texto2[Esp + 1:len(texto2)]  
 Esp = texto2.find(" ")  
print("Em ", texto, " há ", Cont, " palavras")  
print("Em ", texto, " há ", PCT, " palavras começadas e terminadas por", letra)  
palsdif=""  
for i in range(NPDCT):  
 palsdif=palsdif+VetorPal[i]+","  
print("Dessas, ",NPDCT,"Palavras diferentes(",palsdif[0:len(palsdif)-1],")")  
print("Em ", texto, " há ", PCV, " palavras começadas por vogais")  
palsdif=""  
for i in range(NPDCV):  
 palsdif=palsdif+VetorPalCV[i]+","  
print("Dessas, ",NPDCV,"Palavras diferentes(",palsdif[0:len(palsdif)-1],")")

Exercício 4. Dados um conjunto de frases sem sinais de pontuação, terminando com a palavra “fim”, apresente a lista das palavras (precedida da posição na frase), assim como o número de vogais que compõem cada frase. No fim deve escrever o número de palavras encontradas.

Por exemplo, os dados:

DATA “teste de algoritmia”, “quando”, “sera hoje”, “fim” Saída:

1-teste 2-de 3-algoritmia

Em teste de algoritmia há 8 vogais

1-quando

Em quando há 3 vogais

1-sera 2-hoje

Em será hoje há 4 vogais Encontrei 6 palavras.

Algoritmo VariosPedidos

Este algoritmo, dados um conjunto de frases sem sinais de pontuação, terminando com a palavra “fim”,

apresenta a lista das palavras (precedida da posição na frase), assim como o número de vogais que

compõem cada frase; No fim escreve o número de palavras encontradas.

V10 [Inicializar número de palavras]

NP🡨0

V20 [Ler e validar o 1º texto]

Do

Print("Qual o texto?")

Read(texto)

UNTIL texto<>"“

V25 [Estabelecer ciclo para processar textos]

Do V30 TO while texto<>”fim”

V30 [Atualizar texto]

\_\_\_\_texto2<--texto+" “

V40 [Inicializar contador de palavras]

\_\_\_\_Cont🡨0

V50 [Determinar primeiro espaço]

\_\_\_\_Esp<--Index(texto2," ")

V60 [Estabelecer ciclo para processar palavras]

\_\_\_\_Do V700 to B90 while Esp<>0

V70 [Extrair a palavra]

Pal 🡨 SUB(Texto2,1,Esp-1)

V80 [Atualizar contador de palavras]

Cont🡨Cont+1

V90 [Escrever palavra]

Print(Cont,”-”,Pal)

V100 [Atualizar texto]

texto<--Sub(texto2,Esp+1)

V110 [Procurar próximo espaço]

Esp<--Index(texto2," “)

V120 [Atualizar número de palavras]

NP🡨NP+Cont

V130 [Inicializar o número de vogais]

NumV🡨0

V140 [Estabelecer ciclo para processar caracteres]

DO V150 TO V170 FOR I=1 TO LENGTH(Texto)

V150 [Extrair o caracter]

Car🡨 SUB(Texto,I,1)

V160 [Actualizar o número de vogais]

IF INDEX(“aeiou”, Car)>0

THEN NumV🡨NumV+1

V170 [Escrever o número de vogais]

PRINT(“Em “,texto,” há “,NumV,” vogais”)

V180 [Ler e validar o próximo texto]

Do

Print("Qual o texto?")

Read(texto)

UNTIL texto<>"“

V190 [Escrever número de palavras]

PRINT(“Encontrei “,NP, “ Palavras”)

V200 [Terminar]

Exit []

print("Qual o texto?")  
texto=input()  
while len(texto)==0:  
 print("Qual o texto?")  
 texto = input()  
NP=0  
while texto!="fim":  
 #V30 [Atualizar texto]  
 texto2=texto+" "  
 #V40 [Inicializar contador de palavras]  
 Cont=0  
 #V50 [Determinar primeiro espaço]  
 Esp=texto2.find(" ")  
 #V60 [Estabelecer ciclo para processar palavras]  
 while Esp!=-1:  
 Pal = texto2[0:Esp]  
 Cont=Cont+1  
 print(Cont,"-",Pal)  
 texto2 = texto2[Esp + 1:len(texto2)]  
 Esp = texto2.find(" ")  
 NumV = 0  
 NP = NP + Cont  
 for I in range(len(texto)):  
 Car = texto[I]  
 if "aeiou".find(Car) >= 0:  
 NumV = NumV + 1  
 print("Em ", texto, " há ", NumV, " vogais")  
  
 print("Qual o texto?")  
 texto=input()  
 while len(texto)==0:  
 print("Qual o texto?")  
 texto = input()  
print("Encontrei ",NP,"Palavras")

Exercício 5. Pretende-se determinar num texto o número de palavras maiores (alfabeticamente acima) das suas vizinhas.

Por exemplo no texto: “Este e um exemplo de texto a analisar”, há duas dessas palavras: “um”, “texto”

Exercício 6. Dado um texto sem sinais de pontuação:

* Inverta cada uma das palavras do texto.
* Escreva o número de palavras do texto.
* Escreva a frequência de palavras que começam por cada letra do alfabeto.

Entrada:

a casa azul ao lado da casa alta Saída:

O texto com as palavras invertidas = a asac luza oa odal ad asac atla.

Há 8 palavra no texto a casa azul ao lado da casa alta.

Há 4 palavras começadas por a

Há 2 palavras começadas por c

Há 1 palavra começada por d

Há 1 palavra começada por l

Exercício 15. Dado um texto, escreva

* Quantos sinais de pontuação diferentes aparecem no texto - O número de vezes que cada sinal de pontuação ocorre
* Total de sinais no texto.

Por exemplo

Entrada

Este texto, Jorge? Sim. Pode ser? Poder, pode mas não há outro?

Saída

O texto tem 3 sinais de pontuação diferentes.

\*\*\*\*\*\*\*

, ocorre 2 vezes

? ocorre 3 vezes

. ocorre 1 vez

\*\*\*\*\*\*\*

Total de sinais de pontuação = 6

Exercício 7. Dado um texto (em minúsculas e sem sinais de pontuação) escreva esse texto na forma vogais transformadas (em que cada a é transformado em @, e em €, i em !, o em 0 e o u em V). No final deve escrever quantas transformações de letra forma feitas, assim como o número relativo a cada vogal transformada. Por exemplo:

Entrada

um texto para teste de algoritmia

Saída

Texto forma vogais transformadas: Vm t€xt0 p@r@ t€st€ d€ @lg0r!tm!@ Foram efetuadas as seguintes 13 transformações:

@ 4

€ 4

! 2

0 2

V 1

Exercício 8. Dado um texto, não vazio:

Escrever cada um dos números (0 a 9) incluídos no texto no seu correspondente por extenso (um a nove).

Escrever o número de transformações efetuadas

Escrever o número de transformações efetuadas de cada número (0 a 9).

Três exemplos de execução:

|  |  |
| --- | --- |
| Texto? 12123  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Inicial: 12123  Por Extenso: umdoisumdoistres  Foram efetuadas as seguintes 5 transformações:   1. para um 2 2. para dois 2 3. para três 1 |  |
| Texto? ABC 123 321 6 XYZ 2  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Inicial: ABC 123 321 6 XYZ 2  Por Extenso: ABC umdoistrês trêsdoisum seis XYZ dois Foram efetuadas as seguintes 8 transformações:   1. para um 2 2. para dois 3 3. para três 2   6 para seis 1 |  |
| Texto? A10 B11 8C  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Inicial: A10 B11 8C  Por Extenso: Aumzero Bumum oitoC  Foram efetuadas as seguintes 5 transformações:   1. para zero 1 2. para um 3   8 para oito 1 |  |