



UNIVERSIDADE  
PORTUCALENSE

# Algoritmia

**Aula nº 10**  
**2010-2011**



UNIVERSIDADE  
PORTUCALENSE

# Frequência de Letras

**Dado um texto contar o número de vezes que cada letra ocorre.**

**a) Para escrever cada letra?**

# Para escrever cada letra?

## Algoritmo EscreveLetras

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

T10[Ler e validar texto]

DO

PRINT("Texto?")

READ(Tx)

UNTIL Tx<>""

T20 [Determinar o número de caracteres que o texto inclui]

TAM ← LENGTH(Tx)

T30 [Estabelecer quais são as letras]

Alfa ← "abc...z"

T40 [Estabelecer ciclo para processar cada um dos caracteres]

DO T50 TO T60 FOR i=1 to TAM

T50 [Extrair o caracter]

Car ← SUB(Tx, i, 1)

T60 [Escrever o caracter caso seja uma letra]

IF INDEX(Alfa, Car) <> 0

THEN PRINT(Car, " é o nº", INDEX(Alfa, Car), " do alfabeto")

T80 [Terminar]

Exit []



**E para escrever o nº de vezes que cada  
letra aparece?**

**Frase exemplo:  
a casa branca e azul**

- Inicializar vector de frequências**
- Extrair cada um dos 20 caracteres:**
- Verificar se pertence ao alfabeto**
- Incrementar a posição do vector de frqs**
- No fim escrever o conteúdo do vector**

# E para escrever o nº de vezes que cada letra aparece? 2



UNIVERSIDADE  
PORTUCALENSE

Frase exemplo: a casa.

-Inicializar vector de frequências

1	2	3	4	5	6	7	8	...	26
0	0	0	0	0	0	0	0		0

-Extrair caracteres desde o 1º ao último

-1º **a** pertence ao alfabeto? Sim

- Actualizar vector de freqs

1	2	3	4	5	6	7	8	...	26
1	0	0	0	0	0	0	0		0

-2º **" "** pertence ao alfabeto? Não

-3º **c** pertence ao alfabeto? Sim

- Actualizar vector de freqs

1	2	3	4	5	6	7	8	...	26
1	0	1	0	0	0	0	0		0

-4º **a** pertence ao alfabeto? Sim

- Actualizar vector de freqs

1	2	3	4	5	6	7	8	...	26
2	0	1	0	0	0	0	0		0

-5º **s** pertence ao alfabeto? Sim

- Actualizar vector de freqs

1	2	3	4	5	6	...	19	...	26
2	0	1	0	0	0		1		0

- 6º **a** pertence ao alfabeto? Sim

- Actualizar vector de freqs

1	2	3	4	5	6	...	19	...	26
3	0	1	0	0	0		1		0

# Algoritmo Frequência de Letras

Dada uma variável **Texto** alfanumérica determina-se a frequência relativa de cada letra no texto. Usa-se as alfanuméricas **Alfabeto** que contem por ordem todas as letras, **Caracter** que contem a letra corrente em análise e um vector **Frequência[]** que representa a ocorrência de cada letra. **TotalLetras** representa o número total de letras do texto.

# Algoritmo FrqLetras

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 F0 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
    []

```

# Algoritmo PalavrasTxt

**Versão 0:**

**Input:**

**"a arvore da ana e a arvore da joana"**

**output:**

a  
arvore  
da  
ana  
e  
a  
arvore  
da  
joana

"a arvore da ana e a arvore da joana "  
  "arvore da ana e a arvore da joana "  
    "da ana e a arvore da joana "  
      "ana e a arvore da joana "  
        "e a arvore da joana "  
          "a arvore da joana "  
            "arvore da joana "  
              "da joana "  
                "joana "  
                  "



# Algoritmo PalavrasTxt

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 [Determinar a posição do 1º espaço]

Esp←INDEX(Texto, " ")

P40 [Estabelecer ciclo para processar palavras]

DO P50 TO P70 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)

P80 [Terminar]

Exit []

# Algoritmo Frequência palavras

**Versão 1:**

**Input:**

"a arvore da ana e a arvore da joana "

**output:**

"a ocorre 2 vezes"

"arvore ocorre 2 vezes"

"ana ocorre 1 vezes"



# Algoritmo Frequência palavras

## Versão 1:

Texto	Pal	P						Oc					
"a arvore da ana e a arvore da joana "	a	a						1					
		1	2	3	4	5	6	1	2	3	4	5	6
"arvore da ana e a arvore da joana "	arvore	a	arvore					1	1				
		1	2	3	4	5	6	1	2	3	4	5	6
"da ana e a arvore da joana "	da	a	arvore	da				1	1	1			
		1	2	3	4	5	6	1	2	3	4	5	6
"ana e a arvore da joana "	ana	a	arvore	da	ana			1	1	1	1		
		1	2	3	4	5	6	1	2	3	4	5	6
"e a arvore da joana "	e	a	arvore	da	ana	e		1	1	1	1	1	
		1	2	3	4	5	6	1	2	3	4	5	6
"a arvore da joana "	a	a	arvore	da	ana	e		2	1	1	1	1	
		1	2	3	4	5	6	1	2	3	4	5	6
"arvore da joana "	arvore	a	arvore	da	ana	e		2	2	1	1	1	
		1	2	3	4	5	6	1	2	3	4	5	6
"da joana "	da	a	arvore	da	ana	e		2	2	2	1	1	
		1	2	3	4	5	6	1	2	3	4	5	6
"joana "	joana	a	arvore	da	ana	e	joana	2	2	2	1	1	1
		1	2	3	4	5	6	1	2	3	4	5	6

""

### Algoritmo FrequenciaPalavras\_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 [Atualizar 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 [Atualizar 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 [Atualizar 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[]

# Escrever dupletos

**Versão 1:**

**Input: "a arvore da ana"**

**output: a a ar rv vo or re e d da a an na**

**Algoritmo Contador\_Dupletos**

**Este algoritmo, dado um texto, escreve os seus dupletos.**

**C10 [Ler e validar o texto]**

**Do**

**\_\_\_\_Print("Qual o texto?")**

**\_\_\_\_Read(texto)**

**UNTIL texto<>""**

**C20 [Estabelecer ciclo para processar dupletos]**

**Do C30 FOR I=1 to LENGTH(texto)-1**

**C30 [Escrever dupleto]**

**\_\_\_\_PRINT(SUB(texto, I,2))**

**C40 [Terminar]**

**Exit[]**

# Ocorrência de dupletos?

## Versão 2:

**Input: "a arvore da ana"**

**output:**

a 2  
a 2  
ar 1  
rv 1  
vo 1  
or 1  
re 1  
e 1  
d 1  
da 1  
an 1  
na 1

## Algoritmo Contador\_Dupletos\_Diferentes

Este algoritmo, dado um texto, escreve os dupletos diferentes que tem.

C10 [Ler e validar o texto]

Do

\_\_\_\_ Print("Qual o texto?")

\_\_\_\_ Read(texto)

UNTIL texto<>" "

C15 [Inicializar contador de dupletos diferentes]

Cont <-- 0

C20 [Estabelecer ciclo para processar dupletos]

Do C30 FOR I=1 to LENGTH(texto)-1

C30 [Actualizar vector de dupletos]

\_\_\_\_ D[cont+1] <-- SUB(Texto, I,2)

\_\_\_\_ oc[cont+1] <-- 1

\_\_\_\_ qt<-- 1

\_\_\_\_ DO WHILE D[qt]<> SUB(Texto, I,1) and qt<cont

• \_\_\_\_\_ qt <-- qt+1

• \_IF qt<=cont

• \_THEN oc[qt] <-- oc[qt]+1

• \_ELSE cont <-- cont+ 1

C40 [Escreve dupletos diferentes]

DO FOR I=1 to cont

PRINT (D[i], "ocorre ", oc[i], " vezes.")

C50 [Terminar]

Exit[]



# Frequência de letras C

```
#include <stdio.h>
#include <strings.h>
int main ()
{char texto[80];
int ocorrencias[26];
char
alfa[26]="abcdefghijklmnopqrstuvwxyz";
do
{printf("Qual o texto ");
gets(texto);}
while(strcmp(texto,"")!=0);

int i;
for(i=0;i<26;i++)
    ocorrencias[i]=0;

char *pos;
char caracter;
int tam = strlen(texto);
```

```
for(i=0;i<tam;i++)
{ caracter = texto[i];
pos = strchr(alfa, caracter);
if (pos)
    ocorrencias[pos-alfa]++;}

for(i=0;i<26;i++)
    if (ocorrencias[i]!=0)
        printf("A letra %c ocorre %d
vezes \n",alfa[i],ocorrencias[i]);

system ("pause");
}
```

# Frequência palavras C

```

char texto[80];
char palavras[80][20];
int ocorrencias[80];
char * palavra;
char *delimitadores=" ";
int i;
int numeroPalavras=0;

do
{printf("Qual o texto ");
gets(texto);}
while(strcmp(texto,"")!=0) ;

palavra = strtok(texto,
delimitadores);

while (palavra != NULL)
{
    strcpy(palavras[numeroPalavras],palavra);
    i = 0;
    while(strcmp(palavras[i],palavra)!=0)
        i++;
    if(i==numeroPalavras)
    {
        ocorrencias[numeroPalavras]=1;
        numeroPalavras ++;
    }
    else
        ocorrencias[i]++;

    palavra = strtok (NULL, " ");
}
for(i=0;i<numeroPalavras;i ++)
    printf("A palavra %s ocorre %d vezes \n",palavras[i],ocorrencias[i]);

```