



UNIVERSIDADE  
PORTUCALENSE

# Algoritmia

**Aula nº 9**  
**2010-2011**



# Alfanuméricas

**Alfanuméricas, *strings*, são sequências ordenadas sendo cada um membro de um alfabeto**

**Exemplo de alfanuméricas definidas sobre o alfabeto {A,T,C,G}:  
'ATCG', 'ATTCGATCAG', 'GA' ... [constantes alfanuméricas].**

**Uma alfanumérica pode não ter caracteres: alfanumérica vazia.  
"". Não confundir com " " (espaço).**

# Operações

**Concatenação: Escolhem-se dois caracteres e juntam-se para formar uma alfanumérica**

**Exemplo:**

**'Alg' ◦ 'ori' ◦ 'tmia' = 'Algoritmia'**

**A concatenação de duas quaisquer alfanuméricas é sempre uma alfanumérica. Muitas vezes o ◦ aparece como +.**

**Exemplo:**

**Ler e validar o 1º nome**

**Ler e validar o ultimo nome**

**Escrever o nome da pessoa ao estilo Bond, James Bond**

**Terminar**

# Operações

## Algoritmo ConcatenaNome

Este algoritmo dado o primeiro e último nome de uma pessoa, escreve-o sob a forma Ultimo, Primeiro Ultimo.

C10 [Ler e validar o 1º nome]

Do

Print ("Primeiro Nome?")

Read(PrimNome)

Until PrimNome<>""

C20 [Ler e validar o ultimo nome]

Do

Print ("Último Nome?")

Read(UltNome)

Until UltNome<>""

C30 [Escrever o nome da pessoa ao estilo Bond, James Bond]

Print(UltNome," ",PrimNome," ",UltNome)

C40 [Terminar]

Exit []

# Determinar o comprimento de uma alfanumérica



UNIVERSIDADE  
PORTUGALENSE

## **LENGTH (comprimento):**

**Aplicada a uma alfanumérica dá um número inteiro que representa o tamanho ou número de caracteres da alfanumérica.**

**Exemplos de uso:**

**LENGTH('Algoritmia') = 10**

**LENGTH('Aula de Algoritmia') = 18**

# Pesquisar e substituir uma sub-alfanumérica numa alfanumérica

**À operação chama-se sub-alfanumérica, *substring*.  
Tem diversos formatos possíveis.**

Algoritmo: SUB( $a_1$ ,  $a_2$ ,  $a_3$ ) onde:

$a_1$  → alfanumérica de que se pretende extrair uma parte.

$a_2$  → posição onde começa a parte a extrair (O 1º é o 1).

$a_3$  → comprimento dessa parte

SUB('Algoritmia', 8, 3) → 'mia'

SUB('O Porto', 3,2) → 'Po'

SUB('Euro',1,3) → 'Eur'

Exemplo: Dado um nome sob a forma 'Donald Ervin Knuth', transformá-lo em "Knuth, D. E."

### Algoritmo TransformaNome

Este algoritmo dado um nome completo escreve-o na forma  
UltimoNome, PrimeiraLetrado1ºNome. PrimeiraLetrado2ºNome.

```

T10 [Ler e validar nome]
DO
PRINT("Nome?")
READ(Nome)
UNTIL Nome<>""
T20 [Extrair a primeira letra do 1º nome]
Prim←SUB(Nome,1,1)
T30 [Encontrar a posição do 1º espaço]
i←1
DO while SUB(nome,i,1)<>" "
    i←i+1
T40 [Extrair a primeira letra do 2º nome]
Seg←SUB(Nome,i+1,1)
T50 [Encontrar a posição do último espaço]
J=Length(Nome)
DO while SUB(Nome,j,1)<>" "
    J←J-1
T60 [Extrair o último nome]
Ult←SUB(Nome, j+1)
T70 [Escrever o nome no formato pretendido]
PRINT(Ult, ", ", Prim, ". ", Seg, ". ");
T80 [Terminar]
Exit []

```

# INDEX(S,P)

**INDEX(S, P):** Operação de determinar um padrão P numa alfanumérica S onde S representa a alfanumérica (sujeito ou objecto) a examinar para determinar a ocorrência mais à esquerda da sub-alfanumérica P (a padrão). Se P não existir em S o resultado da função é 0.

**Observação:** Por vezes INDEX é designada por MATCH



## Exemplos:



UNIVERSIDADE  
PORTUCALENSE

**INDEX('Donald Ervin Knuth', ' ') dá 7 que é o valor da posição do espaço em branco mais à esquerda.**

**INDEX('Donald Ervin Knuth', 'b') dá 0 pois não há qualquer 'b' em 'Donald Ervin Knuth'.**

**INDEX('Donald Ervin Knuth', 'Knu') dá 14 que é o valor da posição do carácter mais à esquerda de 'Knu'.**

**INDEX('Donald Ervin Knuth', 'Knuh') dá 0 pois não há a subalfanumérica 'Knuh' em 'Donald Ervin Knuth'.**

Exemplo: Dado um nome sob a forma 'Donald Ervin Knuth', transformá-lo em "Knuth, D. E.", versão melhorada



UNIVERSIDADE  
PORTUGALENSE

### Algoritmo TransformaNome

Este algoritmo dado um nome completo escreve-o na forma  
UltimoNome, PrimeiraLetrado1ºNome. PrimeiraLetrado2ºNome.

T10 [Ler e validar nome]

DO

PRINT("Nome?")

READ(Nome)

UNTIL Nome<>""

T20 [Extrair a primeira letra do 1º nome]

Prim←SUB(Nome,1,1)

T30 [Encontrar a posição do 1º espaço]

i←INDEX(Nome, " ")

T40 [Extrair a primeira letra do 2º nome]

Seg←SUB(Nome,i+1,1)

T50 [Encontrar a posição do último espaço]

DO while INDEX(Nome, " ")<>0

Nome=SUB(Nome, INDEX(Nome, " ")+1)

T60 [Extrair o último nome]

Ult←Nome

T70 [Escrever o nome no formato pretendido]

PRINT(ultimo, ", ", Prim, ". ", Seg, ". ");

T80 [Terminar]

Exit []



UNIVERSIDADE  
PORTUCALENSE

## Exemplo a:

**Número de vogais que compõem cada frase.**

## Exemplo a)

### Algoritmo Vogais

Este algoritmo, dado um texto, calcula e escreve o número de vogais que compõem.

V10 [Inicializar o número de vogais]

NumV  $\leftarrow$  0

V20 [Ler e validar o 1º texto]

Do

    Print("Qual o texto?")

    Read(texto)

UNTIL texto <> ""

V30 [Estabelecer ciclo para processar caracteres do texto]

DO V40 TO V50 FOR I=1 TO LENGTH(Texto)

    V40 [Extrair o caracter]

    Car  $\leftarrow$  SUB(Texto, I, 1)

    V50 [Actualizar o número de vogais]

    IF INDEX("aeiou", Car) > 0

        THEN NumV  $\leftarrow$  NumV + 1

V60 [Escrever o número de vogais]

PRINT("Em ", texto, " há ", NumV, " vogais")

V70 [Terminar]

Exit []



UNIVERSIDADE  
PORTUCALENSE

## Exemplo b)

**Dado um texto, contar e escrever as palavras.**

# Exemplo b)

## Algoritmo Palavras

Este algoritmo, dado um texto, conta e escreve as palavras.

P10 [Ler e validar o 1º texto]

Do

Print("Qual o texto?")

Read(texto)

UNTIL texto<>" "

P20 [Atualizar texto]

texto ← texto + " "

P30 [Inicializar contador de palavras]

cont ← 0

P40 [Determinar primeiro espaço]

Esp ← Index(texto, " ")

P50 [Estabelecer ciclo para processar palavras]

Do P60 to P100 while Esp <> 0

P60 [Extrair a palavra]

Pal ← SUB(Texto, 1, Esp-1)

P70 [Atualizar contador de palavras]

Cont ← Cont + 1

P80 [Escrever palavra]

Print(Cont, "-", Pal)

P90 [Atualizar texto]

texto ← SUB(texto, Esp + 1)

P100 [Procurar próximo espaço]

Esp ← Index(texto, " ")

P110 [Escrever o número de palavras]

PRINT("Em ", texto, " há ", Cont, " palavras")

P120 [Terminar]

Exit []

# strlen()

**Em C:**

```
char um[20]="Algoritmia";
char dois[20]="Aula de algoritmia";
int tam1=strlen(um);
int tam2=strlen(dois);
printf ("\n%s tem o comprimento %d",um,tam1);
printf ("\n%s tem o comprimento %d\n",dois,tam2);
```

```
Algoritmia tem o comprimento 10
Aula de algoritmia tem o comprimento 18
Press any key to continue . . . _
```

# strcmp()

```
char Nome[20]="Aula de ";
char v=strcmp("Aula de","Aula de");
printf ("\nstrcmp(Aula de,Aula de)= %d\n", v);
char x=strcmp("Joao","Andre");
printf ("\nstrcmp(Joao,Andre)= %d\n", x);
char z=strcmp("Joao","Ze");
printf ("\nstrcmp(Joao,Ze)= %d\n", z);
```

```
strcmp(Aula de,Aula de)= 0
strcmp(Joao,Andre)= 1
strcmp(Joao,Ze)= -1
Press any key to continue . . . _
```



# Escrever strings

```
char PrimNome[20];
char UltNome[20];
do{
printf ("\nQual o 1º nome?");
scanf("%s",&PrimNome);}
while (strcmp(PrimNome,"")!=0);
do {
printf ("\nQual o ultimoº nome?");
scanf("%s",&UltNome);}
while (strcmp(UltNome,"")!=0);
printf ("\n%s, %s %s\n",UltNome, PrimNome, UltNome);
```

```
Qual o 1º nome?James
Qual o ultimoº nome?Bond
Bond, James Bond
Press any key to continue . . .
```

# strcat()

```
char PrimNome[20];
char UltNome[20];
do{
printf ("\nQual o primeiro nome?");
scanf("%s",&PrimNome);}
while (strcmp(PrimNome,"")==0);
do {
printf ("\nQual o ultimo nome?");
scanf("%s",&UltNome);}
while (strcmp(UltNome,"")==0);
char todoNome[40]="";
strcat(todoNome,UltNome);
strcat(todoNome," ");
strcat(todoNome, PrimNome);
strcat(todoNome," ");
strcat(todoNome, UltNome);
printf("\nNome Novo=%s\n",todoNome);
```

```
Qual o primeiro nome?James
Qual o ultimo nome?Bond
Nome Novo=Bond, James Bond
Press any key to continue . . .
```

# strcpy()

```
char nome[20] = "Jose Santos";
char v[20];
strcpy(v,nome);
printf("Copiei o nome(%s) para v(%s)", nome, v);
```

```
Copiei o nome(Jose Santos) para v(Jose Santos)P:
```

**Não posso fazer v=nome;**

## strstr()

```
char um[20]="algoritmia";
char dois[20]="Aula de algoritmia";
char onde=strstr(dois,um);
printf ("\n%s em %s aparece no %c",um, dois, onde);
char onde2=strstr( um, dois);
if (onde2!=0)
printf ("\n%s em %s aparece no %d\n",dois, um,onde2);
else
printf ("\n%s em %s Não aparece \n",dois, um);
```

```
algoritmia em Aula de algoritmia aparece no 8
Aula de algoritmia em algoritmia Não aparece
Press any key to continue . . . _
```

## strchr()

```
char s[20] = "aula de algoritmia";
char um='g';
char *pos = strchr(s, um);
if (pos)
    printf("%c encontrado em %s na posição %d.\n",um, s, pos - s+1);
else
    printf("%c não encontrado em %s.\n",um, s);
char dois='z';
pos= strchr(s, dois);
if (pos)
    printf(" %c encontrado em %s na posição %d.\n",dois, s, pos - s+1);
else
    printf("%c não encontrado em %s.\n",dois, s);
```

```
Character g encontrado em aula de algoritmia na posição 11.
Character z não encontrado em aula de algoritmia.
Press any key to continue . . .
```

# strtok ()

```
char texto[] = "Um exemplo? Com caracteres, sinais ou outros!?!";
char * pch;
char nova[80] = "";
char * sinais = ",!?.;";
printf ("String inicial: %s\n", texto);
pch = strtok (texto, sinais);
while (pch != NULL)
{ strcat(nova, pch);
  pch = strtok (NULL, sinais); }
printf ("String nova:  %s\n", nova);
```

```
String inicial: Um exemplo? Com caracteres, sinais ou outros!?!
String nova:    Um exemplo Com caracteres sinais ou outros
Press any key to continue . . . _
```

# Contar vogais

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

```
int numVogais=0;
char * vogais="aeiou";
char *pos;
char caracter;
int tam = strlen(texto);
int i;
for(i=0;i<tam;i++)
{ caracter = texto[i];
  pos = strchr(vogais, caracter);
  if (pos)
    numVogais++;
}
printf("No texto %s ha %d vogais. ",texto, numVogais);
system ("pause");
}
```

```
Qual o texto bom dia de sol
No texto bom dia de sol ha 5 vogais. Press any
```

# Contar palavras

```
#include <stdio.h>
#include <strings.h>
int main ()
{char texto[80];
do
{printf("Qual o texto ");
gets(texto);}
while(strcmp(texto,"")!=0) ;
char texto2[80];
strcpy(texto2,texto);
int numeroPalavras=0;
char * palavra;
palavra = strtok(texto, " ");
while (palavra != NULL)
{numeroPalavras++;
palavra = strtok (NULL, " "); }
printf("No texto %s ha %d palavras. ",texto2, numeroPalavras);
system ("pause");
}
```

Qual o texto bom dia de sol  
No texto bom dia de sol ha 4 palavras. Press