

Fundamentos de Programação 1

Estrutura de Dados Homogêneas

**“vetores de números
e de caracteres”**

Slides C – Número 4

Prof. SIMÃO

Vetores de números e de caracteres (*string*)

Vetores de caracteres são “especiais”.

Um vetor comporta uma ‘frase’, chamada no jargão da informática de *string*.

Na verdade, nesse jargão, o vetor de caracteres é chamado (ele mesmo) de *string*.

Exercício

Elabore um programa que recebe informações de 30 alunos, sendo estas informações o nome e as duas notas de cada um.

O programa deve ainda fornecer a média de notas de cada aluno apresentando seu nome.

Solução para o exercício – v1

```
#include "stdio.h"
//Programa de Nomes e Notas de Alunos.
int main ( )
{
    float notas[3];
    char nome[150]; // Um vetor de caracteres ou string.
    int cont;
    for ( cont = 0; cont < 30; cont = cont + 1 )
    {
        fflush ( stdin );

        printf ( "\n Digite o nome do aluno: \n" );

        int cont2;
        for ( cont2 = 0; cont2 < 149; cont2 = cont2 + 1 )
        {
            scanf ( "%c", &nome [cont2] );
            if ( '\n' == nome[cont2] )
            {
                break;
            }
        }
        nome [cont2] = '\0'

        printf ( "Digite as duas notas do aluno \n" );
        printf ( "Primeira nota: " );
        scanf ( "%f", &notas[0] );
        printf ( "Segunda nota: " );
        scanf ( "%f", &notas[1] );
        // não esquecer de validar as variáveis...

        notas[2] = ( notas[0] + notas[1] ) / 2.0;
```

```
printf ( " \n A média do aluno " );
int cont2;
for ( cont2 = 0; cont2 < 150; cont2 = cont2 + 1 )
{
    printf ("%c", nome[cont2]);
    if ( '\0' == nome[cont2] )
    {
        break;
    }
}
printf ( " é %f: \n", notas[2] );
}
return 0;
}
```

Mostra
o
nome

Leitura
do
nome

Todo vetor de caracteres é finalizado com um caracter 'especial', o caracter '/0'.

Solução para o exercício – v.2

```
// VetorCaracteres2.cpp : Defines the entry point for the console application.
```

```
#include "stdio.h"
```

```
//Programa Nomes e notas de alunos.
```

```
int main ( )
```

```
{
```

```
float notas[3];
```

```
char nome[150]; // Um vetor de caracteres ou "string".
```

```
int cont;
```

```
for ( cont = 0; cont < 30; cont = cont + 1 )
```

```
{
```

```
fflush ( stdin );
```

```
printf ( "\n Digite o nome do aluno: \n" );
```

```
scanf ( "%s", nome );
```

```
printf ( "Digite as duas notas do aluno \n" );
```

```
printf ( "Primeira nota: " );
```

```
scanf ( "%f", &notas[0] );
```

```
printf ( "Segunda nota: " );
```

```
scanf ( "%f", &notas[1] );
```

```
// não esquecer de validar as variáveis...
```

```
notas[2] = ( notas[0] + notas[1] ) / 2.0;
```

```
printf ( " \n A média do aluno %s é %f: \n", nome, notas[2] );
```

```
}
```

```
return 0;
```

```
}
```

Leitura do nome
via %s no scanf().

scanf() para com
espaço

Mostra o nome

O %s adiciona o caracter especial '\0'
automaticamente ao vetor de caracteres
(i.e. a string)

Solução para o exercício – v.3

```
// VetorCaracteres3.cpp : Defines the entry point for the console application.
```

```
#include "stdio.h"  
#include "string.h"
```

```
// Programa Nomes e notas de alunos.
```

```
int main ( )  
{  
    float    notas [ 3 ];  
    char     nome [ 150 ];    // Um vetor de caracteres ou string.  
    int      cont;  
    for ( cont = 0; cont < 2; cont = cont + 1 )  
    {  
        fflush( stdin );  
        printf ( "\n Digite o nome do aluno: \n" );  
        gets ( nome );  
  
        printf ( "Digite as duas notas do aluno. \n" );  
        printf ( "Primeira nota: " );  
        scanf ( "%f", &notas[0] );  
        printf ( "Segunda nota: " );  
        scanf ( "%f", &notas[1] );  
        // não esquecer de validar as variáveis...  
  
        notas[2] = ( notas[0] + notas[1] ) / 2.0;  
  
        printf ( " \n A média do aluno %s é %f: \n", nome, notas[2] );  
    }  
  
    return 0;  
}
```

gets ()

Leitura do nome
via gets ().

Solução para o exercício – v.4

```
#include "stdio.h"
#include "string.h"

// Programa Nomes e notas de alunos.
int main ( )
{
    float    notas[3];
    char     nome[150];    // Um vetor de caracteres ou string.
    int cont;
    for ( cont = 0; cont < 2; cont = cont + 1)
    {
        fflush(stdin);

        puts ( "\n Digite o nome do aluno: \n" );
        gets ( nome );

        puts ( "Digite as duas notas do aluno \n" );
        puts ( "Primeira nota: " );
        scanf ( "%f", &notas[0] );
        puts ( "Segunda nota: " );
        scanf ( "%f", &notas[1] );
        // não esquecer de validar as variáveis...

        notas[2] = ( notas[0] + notas[1] ) / 2.0;
        printf ( " \n A média do aluno %s é %f: \n", nome, notas[2] );
    }

    return 0;
}
```

puts ()

Não esquecer da validação... *vide* algoritmo abaixo.

Algoritmo 'Nomes e notas de alunos'.

Início

Real notas[3];

Caracter nome[150];

Inteiro cont;

para cont de 0 até 29 passo 1 faça

 Imprima ("Digite o nome do aluno: ");

 Leia (nome);

faça

 Imprima ("Digite a primeira nota do aluno");

 Leia (nota[0]);

se ((nota[0] < 0) **ou** (nota[0] > 10)) **então**

 Imprima ("Nota não válida.");

fim-se

enquanto ((nota[0] < 0) **ou** (nota[0] > 10))

faça

 Imprima ("Digite a primeira nota do aluno");

 Leia (nota[1]);

se ((nota[1] < 0) **ou** (nota[1] > 10)) **então**

 Imprima ("Nota não válida.");

fim-se

enquanto ((nota[1] < 0) **ou** (nota[1] > 10))

 notas[2] ← notas[0] + notas[1];

 Imprima (" A média do aluno %s é %i: ", nome, notas[2]);

fim-para

Fim.

ou ainda...

Algoritmo ‘Nomes e notas de alunos’.

Início

Real notas[3];

Caracter nome[150];

Inteiro cont;

para cont **de** 0 **até** 29 **passo** 1 **faça**

Imprima (“Digite o nome do aluno: ”);

Leia (nome);

repita

Imprima (“Digite a primeira nota do aluno”);

Leia (nota[0]);

se ((nota[0] < 0) **ou** (nota[0] > 10)) **então**

Imprima (“Nota não válida.”);

fim-se

até ((nota[0] >= 0) **e** (nota[0] <= 10))

repita

Imprima (“Digite a segunda nota do aluno”);

Leia (nota[1]);

se ((nota[1] < 0) **ou** (nota[1] > 10)) **então**

Imprima (“Nota não válida.”);

fim-se

até ((nota[1] >= 0) **e** (nota[1] <= 10))

 notas[2] ← notas[0] + notas[1];

Imprima (“ A média do aluno %s é %i: ”, nome, notas[2]);

fim-para

Fim.

Caracter de finalização - 1

```
#include <stdio.h>
#include <string.h>
int main()
{
    int numero = 0;    int espacos = 0;
    char nome[100];

    printf ("Digite um nome e em seguida digite '! \n");
    fflush (stdin);
    gets (nome);

    for (int cont = 0; cont < 99; cont++)
    {
        if ( nome[cont] == ' ')
        {
            espacos = espacos + 1;
        }

        if(nome[cont] == '!')
        {
            numero = cont - espacos;
            break;
        }
    }

    printf("O numero de caracteres do nome digitado eh %i \n", numero);
    getchar();
    return 0;
}
```

Caracter de finalização - 2

```
#include <stdio.h>
#include <string.h>
int main()
{
    int numero = 0;    int espacos = 0;
    char nome[100];

    printf ("Digite um nome : \n");
    fflush (stdin);
    gets (nome);

    for (int cont = 0; cont < 99; cont++)
    {
        if ( nome[cont] == ' ')
        {
            espacos = espacos + 1;
        }

        if(nome[cont] == '\0')
        {
            numero = cont - espacos;
            break;
        }
    }

    printf("O numero de caracteres do nome digitado eh %i \n", numero);
    getchar();
    return 0;
}
```

Exercício

Elabore um programa para ler o nome e o sobrenome de dez pessoas, imprimindo-os no seguinte formato: *sobrenome, nome*.

-Faça uma primeira versão do algoritmo lendo cada *frase* de uma única vez.

- Faça uma segunda versão do algoritmo lendo caracter a caracter de cada frase..

Obs: *frase* significa um vetor de caracteres...

Exercícios Complementários

a) **Elabore um programa que leia uma *frase* (vetor de caracteres) de no máximo 100 caracteres e conte quantos caracteres a frase possui efetivamente.**

- **Obs.: Vamos considerar que toda *frase* (também chamada de *string*) é finalizado pelo caracter “\0”**

b) **Elabore um programa que leia uma *string* e conte quantas vogais há nela.**

c) **Elabore um programa que leia uma *string* de no máximo 200 caracteres. O programa deve inserir um caracter escolhido pelo usuário em uma posição do vetor, também escolhida pelo usuário. O algoritmo deve utilizar dois vetores: o original e o modificado.**