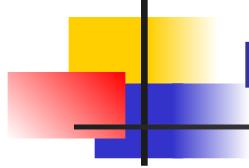


Strings

- Vetores de Caracteres -





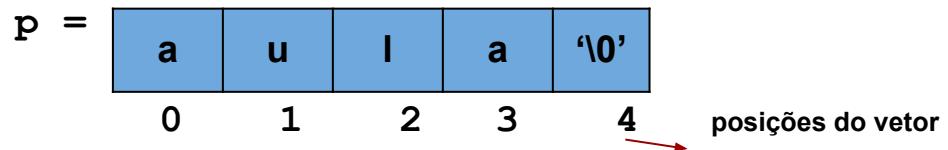
Roteiro

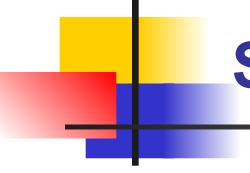
- Strings
- Leitura de String
- Exibição de String
- Inicialização de String
- Exemplos de Manipulação de Strings
- Algumas funções da biblioteca string.h

Strings, cadeias ou sequências de caracteres

- Strings são vetores de caracteres, vetores do tipo `char`.
- Uma palavra é armazenada, na Linguagem C, como um vetor do tipo `char`
- Toda string é terminada em nulo. Nulo é representado por '`\0`'.
- Por conter este terminador, vetores de caracteres são **declarados** com um caractere a mais, para armazenar o '`\0`' .
 - Para armazenarmos a palavra "aula", por exemplo, podemos declarar a *string* da seguinte forma:

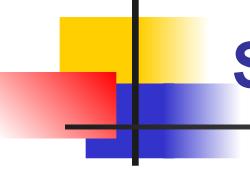
```
char p[5];
```





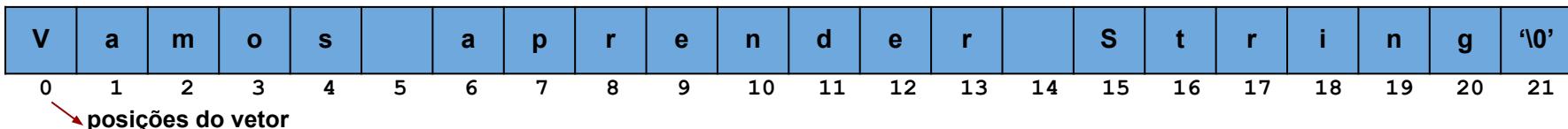
Strings, cadeias ou sequências de caracteres

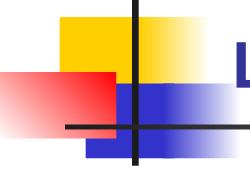
- Podemos também armazenar uma frase, por meio de uma String em C, por exemplo para armazenar a frase “Vamos aprender String”:
 - Qual deve ser o tamanho da string?
 - O caracter espaço também ocupa uma posição no vetor de char.



Strings, cadeias ou sequências de caracteres

- Podemos também armazenar uma frase, por meio de uma String em C, por exemplo para armazenar a frase “Vamos aprender String”:
 - Qual deve ser o tamanho da string?
 - O caracter espaço também ocupa uma posição no vetor de char.

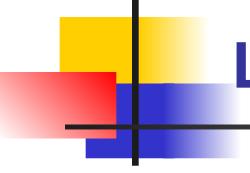




Leitura de String

- Para lermos uma string, podemos usar a função `scanf`
- Na leitura de strings com `scanf`, não se deve colocar o símbolo `&` antes do nome da variável, pois o próprio nome da string contém o endereço.
 - Exemplo de leitura:

```
char nome[21];  
  
scanf("%s", nome);
```

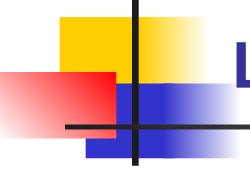


Leitura de String

- A biblioteca `string.h` possui funções para manipulação de strings, na linguagem C.
- Para a leitura de strings, há a função `gets()`
 - Exemplo de leitura com `gets()`:

```
char nome[21];  
  
gets(nome);  
  
○ É preciso incluir a biblioteca string.h:
```

```
#include <string.h>
```



Leitura de String

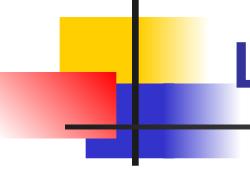
- A biblioteca `string.h` possui funções para manipulação de strings, na linguagem C.
- Para a leitura de strings, há a função `gets()`
 - Exemplo de leitura com `gets()`:

```
char nome[21];  
  
gets(nome);
```

- É preciso incluir a biblioteca `string.h`:

```
#include <string.h>
```

```
#include <stdio.h>  
#include <strings.h>  
  
int main(){  
    char nome[21];  
  
    printf("Digite seu nome: ");  
    gets(nome);  
  
    printf("Boa noite %s", nome);  
    return 0;  
}
```



Leitura de String

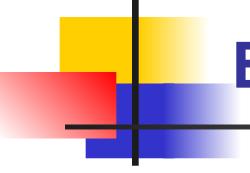
- Podemos também fazer a leitura da string com a função fgets()

```
#include <stdio.h>
#include <strings.h>

int main(){
    char nome[21];

    printf("Digite seu nome: ");
    fgets(nome, sizeof(nome), stdin);

    printf("Boa noite %s", nome);
    return 0;
}
```



Exibição de String

- Para exibir string, podemos usar
 - a função `printf` com a diretiva `%s`, como no exemplo anterior

ou

- com a função `puts()`, neste caso, precisamos incluir o cabeçalho `#include <string.h>`
 - `puts(nome)`

```
#include <stdio.h>
#include <strings.h>

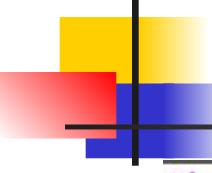
int main(){
    char nome[21];

    printf("Digite seu nome:");
    scanf("%s", nome);

    puts(nome);
    return 0;
}
```



Exemplos de Manipulação de Strings



Vamos testar este exemplo?

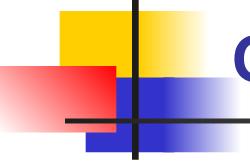
```
#include <stdio.h>

int main() {
    char palavra[20];
    int i;

    printf("Digite UMA palavra qualquer: ");
    scanf("%s", palavra);

    printf("Palavra escrita verticalmente: \n");
    for(i=0; palavra[i] != '\0'; i++){
        printf("%c\n", palavra[i]);
    }

    printf("Palavra digitada: %s\n", palavra);
}
```



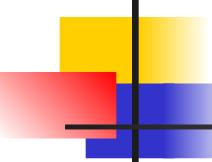
O que este programa imprimirá na tela?

```
#include <stdio.h>

int main() {
    char a[20];
    int i=0;

    printf("Digite UMA palavra: ");
    scanf("%s", a);

    while(a[i]!='\0'){
        printf("%c\n", a[i]);
        i++;
    }
    return 0;
}
```



Contar a quantidade de caracteres de uma palavra:

```
#include<stdio.h>

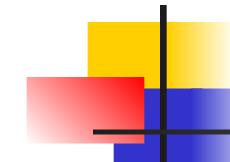
int main() {
    int i;
    char string[20];

    printf("Digite UMA palavra: ");
    scanf("%s",string);

    for(i=0; string[i]; i++);
    //este laço equivale ao laço abaixo:
    //for(i=0; string[i]!='\0'; i++);

    printf("\nA palavra %s tem %d letras", string, i);
}
```

- O '\0' é o terminador da string



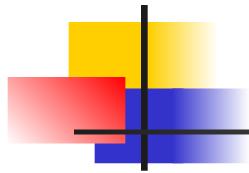
Vamos truncar uma string, usando '\0'

```
#include<stdio.h>

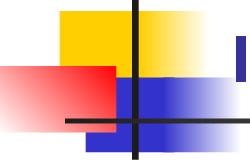
int main() {
    char str[20];
    int i;

    scanf("%s", str);

    for(i=0; str[i]; i++){
        if (str[i]=='a') {
            str[i] = '\0';
            printf ("Palavra cortada no primeiro 'a': %s\n", str);
        }
    }
    return 0;
}
```



Algumas funções da biblioteca string.h



Incluir o cabeçalho #include <string.h>

Nome	Função
<u>strcpy</u> (s1,s2)	Copia s2 para s1
<u>strcat</u> (s1,s2)	Concatena s2 ao final de s1
<u>strlen</u> (s1,s2)	Retorna o tamanho de s1
<u>strcmp</u> (s1,s2)	Retorna 0 (falso) se s1=s2; menos que 0 se s1<s2; maior que zero se s1>s2.
<u>strchr</u> (s1,ch)	Retorna um ponteiro para a primeira ocorrência de ch em s1
<u>strstr</u> (s1,s2)	Retorna um ponteiro para a primeira ocorrência de s2 em s1