

# Strings

Prof. Karl Apaza Agüero

# Strings - Sequências de caracteres

- ▶ Uma *string* é um vetor de caracteres alfanuméricos (i.e. letras, números e símbolos)
  - ▶ Exemplo de string: “abcdefhi12!” (uma senha de 8 letras, 2 números e 1 caractere especial)
- ▶ A *string* deve ser terminada por um caractere nulo ‘\0’ (final de string)
  - ▶ Deve-se sempre reservar um espaço para este caractere
    - ▶ Por exemplo, para armazenar uma cadeia de 40 caracteres deve-se reservar um vetor de 41 de caracteres
- ▶ Uma *string* nula ou vazia é representada por “”
- ▶ Uma string com um único caractere é diferente do caractere de tipo char

```
"Z" ≠ 'Z'
"3" ≠ '3'
```

# Strings

## ► Declaração

```
char <nome da string> [Tamanho];
```

## ► Exemplo

```
char senha[12]; //reservar um espaço para o caractere nulo
```

## ► Inicialização

```
char oi[3];  
oi[0]='\o';  
oi[1]='\i';  
oi[2]='\0'; // caractere nulo
```

## ► Declaração e Inicialização

```
char texto1[6] = {'t', 'e', 'x', 't', 'o'};  
char texto2[] = "texto";
```

```
texto1[1] = '3'; // Alterando uma das posições da string
```

# Strings

## ► Entradas/saídas com strings

```
// Leitura de uma string a partir do teclado  
// scanf ou cin param com espaço, tab ou enter  
cin>>texto;  
scanf("%s ", texto);
```



**Sem o '&' no caso de strings**

```
// Apresentação da string  
// printf ou cout param no caractere nulo '\\0'  
cout<<texto;  
printf("%s", texto);
```

# Strings

- ▶ Exemplo: criar e imprimir a *string* “Rio”

```
#include <stdio.h>
int main ()
{
    char cidade[4];
    cidade[0] = 'R';
    cidade[1] = 'i';
    cidade[2] = 'o';
    cidade[3] = '\0';
    printf("%s\n", cidade);
    return 0;
}
```

} `char cidade[] = {'R', 'i', 'o'};`

`char cidade[] = "Rio";`

# Strings

- ▶ Biblioteca para manipulação de cadeias de caracteres: `string.h`

Nome da função	Propósito
<code>strlen(texto)</code>	Retorna o tamanho da <i>string</i> texto em número de caracteres
...	...

- ▶ Exemplo
  - ▶ Saída: 5

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

int main ()
{
    char nome[] = "texto";
    //strlen é uma função definida em string.h
    int tam = strlen(nome);
    printf("%d\n", tam);
    return 0;
}
```

# Strings

## ► Problema:

Fazer um programa que calcule o tamanho de uma cadeia de caracteres, sem espaços, informada pelo usuário (sem usar a função `strlen`). Tamanho máximo da cadeia = 100.

# Strings

```
#include <stdio.h>
int main(){

    char str[101];
    scanf("%s",str);
    int i=0;
    while(str[i]!='\0')
        i++;

    printf("%d\n",i);

    return 0;
}
```



# Strings

## ► Problema:

Imprimir uma sub-string. Fazer um programa que solicite ao usuário uma cadeia de caracteres, o valor de uma posição (a partir de 0) e um valor referente a uma quantidade de caracteres, e imprima o pedaço da cadeia de caracteres resultante. Tamanho máximo da cadeia = 20.

E.g.: Consideremos a *string* “Texto”. Digamos que o usuário deseja imprimir a *sub-string* “ext”. Para tanto, ele deve informar, além da string original, a posição 1 e a quantidade 3.

### **Entrada**

Texto

1

3

### **Saída**

ext

# Strings

```
#include <stdio.h>
int main(){

    char str[21];
    scanf("%s",str);
    int p,q;
    scanf("%d%d",&p,&q);

    for(int i=p; (i<p+q)&&(str[i]!='\0'); i++)
        printf("%c",str[i]);
    printf("\n");

    return 0;
}
```

# Strings

## ► Problema:

Fazer um programa que solicite ao usuário uma cadeia de caracteres (sem espaços) e indique se a string informada é ou não um palíndromo (i.e. se o inverso da cadeia é igual a ela). Tamanho máximo da cadeia = 100.

E.g.: Considere as *entradas (palíndromos)*:

1. "anilina"
2. "omitoeotimo"

# Strings

```
#include <stdio.h>
int main() {

    char str[101];
    scanf("%s", str);

    int i=0;
    while(str[i]!='\0')
        i++;

    int meio=i/2, pal=1;
    for(int j=0; j<meio; j++)
        if(str[j]!=str[i-j-1]){
            pal=0;
            break;
        }

    if(pal) printf("Palindromo\n");
    else    printf("Nao e' palindromo\n");

    return 0;
}
```