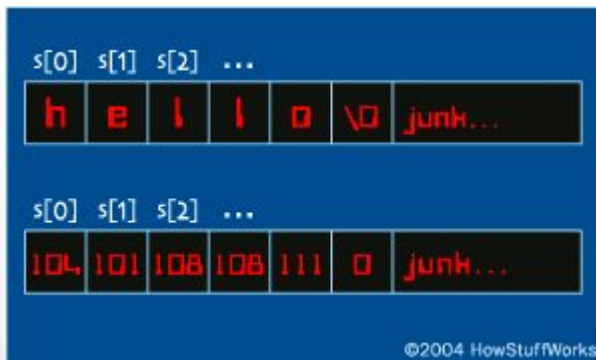


Strings

Sumário

- Definição
- Uso
- Operações básicas sobre Strings



O que é uma String?

- Representa um texto (vários caracteres).
- Logo, como são vários caracteres, usamos um **vetor de char**.
- Terminado com um caractere especial: **'\0'**.

Exemplo

```
char palavra[50];
```

'S'	'T'	'R'	'I'	'N'	'G'	'\0'
-----	-----	-----	-----	-----	-----	------

'S'	'T'	'R'	'\0'	'I'	'N'	'G'
-----	-----	-----	------	-----	-----	-----

Tratamento Especial!

- Em outras linguagens, Strings são tipos primitivos
- Em C não é, mas é tratada de forma diferenciada em algumas funções.

Leitura e escrita

```
printf(“%s\n”, string);
```

```
scanf(“%s”, string);
```

- As funções de leitura e escrita tratam a string em função do seu ‘\0’.

Outras formas de ler

Quando usamos `scanf("%s")`, a leitura será feita até **um espaço em branco**!

Ex: “Duas palavras”

```
scanf("%s", texto);  
printf("%s", texto);
```

“Duas”

Função gets

- A função gets permite que você leia uma linha inteira, terminando a leitura apenas com um \n

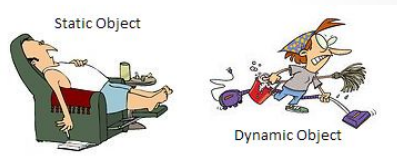
Ex: “Duas palavras”

```
    gets(texto);  
    printf(“%s”, texto);
```

“Duas palavras”

Definindo tamanhos

- Como todo vetor, é necessário que se aloque o tamanho antes de usar.



- Lembre-se de contar +1 para o caractere '\0'!
- Alocação dinâmica será vista posteriormente, ao final dessa disciplina ou em LP1.

Definindo tamanho

- Pode-se colocar o tamanho direto na definição

```
char texto[100]
```

- Ou definir uma constante que guarda o valor

```
#define N 100  
char texto[N];
```

Operações com Strings

- Existe uma série de operações que são importantes quando fazemos aplicações com strings
 - Todas as linguagens têm funções definidas para essas operações.

Tamanho da String

- Dada uma String qualquer, descobrir quantos caracteres ela tem.

Solução

```
int tamanhoString(char string[]){  
    int i = 0;  
    while(string[i] != '\0'){  
        i++;  
    }  
    return i;  
}
```

```
int tamanhoString(char string[]){  
    int i;  
    for (i = 0; string[i] != '\0'; i++);  
    return i;  
}
```



Copiar Strings

- Dada uma String `a`, copiar o seu valor para outra String.

Solução

```
void copiaString(char st1[], char st2[]){  
    int i;  
  
    for(i=0; st1[i] != '\0'; i++){  
        st2[i] = st1[i];  
    }  
  
    st2[i] = '\0';  
}
```

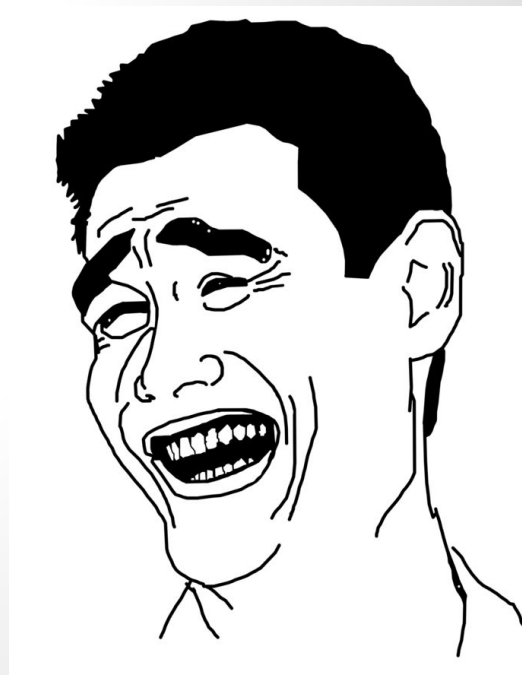


Concatenação de Strings

- Dadas duas Strings, a e b, formar uma única String ab (juntar as duas em uma só, a na frente, b atrás)

Solução

```
void juntaString(char st1[], char st2[], char cat[]){  
    int t1 = tamanhoString(st1);  
    int t2 = tamanhoString(st2);  
    int i, j;  
    for(i=0; i<t1; i++) {  
        cat[i] = st1[i];  
    }  
    for(j=0; j<t2; j++, i++) {  
        cat[i] = st2[j];  
    }  
    cat[i] = '\\0';  
}
```



Achar Substring

- Dadas duas strings, s1 e s2, saber se s2 é substring de s1. Ou seja, se s1 contém s2.
- Deve retornar o índice do início de s2 em s1 ou -1 se s2 não for uma subtring de s1

Ex: s1= "ABRACADABRA" e s2="CADA" \rightarrow 4

Solução

```
int substring(char st[], char subst[]){  
    int t1 = tamanhoString(st);  
    int t2 = tamanhoString(subst);  
    int i, j;  
    for(i=0; i <= t1 - t2; i++) {  
        for (j=0; j < t2 && st[i+j] == subst[j]; j++);  
        if (j == t2) {  
            return i;  
        }  
    }  
    return -1;  
}
```

Está tudo pronto...

- Biblioteca string.h
 - strlen
 - strcpy
 - strcat
 - strstr
- E outras funções!
 - strcmp
 - memset
 - ...



Inicializando Vetores, Strings

- Uma função bastante útil da biblioteca de strings é a função `memset`!
 - `memset(array, valor, tamanho);`
- Ela inicializa uma área de memória com o valor escolhido pelo programador

EX:

```
memset(vetor, 0, sizeof(int)*100);  
memset(palavra, 0, sizeof(palavra));
```

Outras funções - escrita

- **puts** - imprime uma string, com \n no final.
- **sprintf** - realiza a impressão em uma string, em vez de na entrada padrão
- **fprintf** - realiza a impressão em um arquivo
- **putc** - imprime um caractere em um stream
- **putchar** - imprime um caractere na saída padrão

Outras funções - leitura

- **sscanf** - realiza um scanf em uma string, em vez de na entrada padrão
- **fscanf** - lê de um arquivo
- **getc** - lê um character do stream indicado
- **getchar** - lê um character da entrada padrão
- **ungetc** - “devolve” um character para o stream de leitura

Atividades

- Escrever uma função que, dado uma string S e um caracter C , retorna a quantidade de vezes que C aparece em S .
 - Ex: “ABRACADABRA”, “A” \rightarrow 5
- Escrever uma função para verificar se uma palavra é um palíndromo.
 - Ex: “OSSO” \rightarrow 1
 - Ex2: “OSSADA” \rightarrow 0