

IP Strings

Hebert Coelho

Instituto de Informática
Universidade Federal de Goiás

Roteiro

- Exercícios - vetores

Roteiro

- Exercícios - vetores
- Strings

Roteiro

- Exercícios - vetores
- Strings
- Exemplos

Ordenação de vetores

Insertion Sort: percorre um vetor de elementos da esquerda para a direita. A medida que avança vai deixando os elementos mais à esquerda ordenados. O algoritmo de inserção funciona da mesma maneira com que muitas pessoas ordenam cartas em um jogo de baralho como o pôquer.

Insertion sort

```
int main(){
    int i, j, chave, vet[10];
    for(j = 0; j < 10; j++){
        scanf("%d", &vet[j]);
    }
    for(j=1; j<10; j++){
        chave = vet[j];
        i = j - 1;
        while(i >= 0 && vet[i] > chave){
            vet[i + 1] = vet[i];
            i --;
        }
        vet[i + 1] = chave;
    }
    return 0;
}
```

String

- Uma string é um vetor de caracteres terminado com um caractere nulo.

String

- Uma string é um vetor de caracteres terminado com um caractere nulo.
- Declaração de uma string
`char nome_da_string[tamanho];`

String

- Uma string é um vetor de caracteres terminado com um caractere nulo.
- Declaração de uma string
`char nome_da_string[tamanho];`

Exemplo:

j	o	a	o	\0	...	
---	---	---	---	----	-----	--

Entrada e saída de strings

- A função `gets()` permite ler uma string digitada pelo usuário. A função `gets()` coloca o terminador nulo na string, quando você aperta a tecla “Enter”.

Entrada e saída de strings

- A função **gets()** permite ler uma string digitada pelo usuário. A função **gets()** coloca o terminador nulo na string, quando você aperta a tecla “Enter”.
- A função **puts()** permite escrever uma string na tela.

Exemplo:

```
#include <stdio.h>
int main (){
    char nome[100];
    printf("Digite uma string: ");
    gets(nome);
    puts(nome);
    return 0;
}
```

Entrada e saída de strings

Pode-se usar também a função `scanf` e `printf`.

Exemplo:

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

Entrada e saída de strings

gets não é uma função segura. Por quê? Simplesmente porque com `gets` pode ocorrer um estouro da quantidade de posições que foi especificada na string.

Funções para manipulação de strings

strcpy

Sua forma geral é:

```
strcpy (string_destino, string_origem);
```

A função **strcpy()** copia a string-origem para a string- destino.

Funções para manipulação de strings

strcat

A função **strcat()** tem a seguinte forma geral:

```
strcat (string_destino, string_origem);
```

A string de origem permanecerá inalterada e será anexada ao fim da string de destino.

Exercícios

- 1 - Crie um programa que leia uma string do teclado e verifique o seu tamanho. (5 min)
- 2 - Crie um programa que leia duas strings do teclado e verifiquem se as strings são iguais. (5min)

Funções para manipulação de strings

strlen

Sua forma geral é:

```
strlen (string);
```

A função **strlen()** retorna o comprimento da string fornecida. O terminador nulo não é contado.

Funções para manipulação de strings

strcmp

Sua forma geral é:

```
strcmp (string1,string2);
```

A função **strcmp()** compara a string 1 com a string 2. Se as duas forem idênticas a função retorna zero. Se elas forem diferentes a função retorna não-zero.

Exercícios

3 - Crie um programa que verifique se uma senha foi digitada corretamente. Caso a senha digitada não seja correta, escreva a mensagem "senha inválida" e solicite ao usuário para tentar até que a senha seja correta. Obs.: A senha é "pc01".