

Universidade Federal do Ceará - Campus Quixadá  
QXD0010 – Estrutura de Dados  
Prof. Atílio Gomes Luiz

## PONTEIROS E ALOCAÇÃO DINÂMICA DE MEMÓRIA

1. Por que o código abaixo está errado?

```
1 void troca (int *i, int *j) {  
2     int *temp;  
3     *temp = *i;  
4     *i = *j;  
5     *j = *temp;  
6 }
```

2. Seja  $v$  um vetor de `int`. Suponha que cada `int` ocupa 8 bytes no seu computador. Se o endereço de  $v[0]$  é 55000, qual o valor da expressão  $v + 3$ ?

3. Escreva uma função `mm` que receba um vetor inteiro  $A$  com  $n$  elementos e os endereços de duas variáveis inteiras, digamos `min` e `max`, e deposite nestas variáveis o valor de um elemento mínimo e o valor de um elemento máximo do vetor. A função `mm` deve seguir o seguinte protótipo: `void mm(int A[], int n, int *min, int *max);`

Assim como na questão anterior, nesta questão, escreva uma função `main` que use a função `mm`.

4. Faça uma função `MAX` que recebe como entrada um inteiro  $n$ , uma matriz inteira  $A_{n \times n}$  e devolve três inteiros:  $k$ ,  $l$  e  $c$ , tal que

- $k$  é o maior elemento de  $A$  e é igual a  $A[l][c]$ .

Se o elemento máximo ocorrer mais de uma vez, indique em  $l$  e  $c$  qualquer uma das possíveis posições. Use ponteiros para os argumentos.

Escreva uma função `main` que use a função `MAX`.

5. Implemente a seguinte função usando ponteiros:

```
char *strcpy(char *destino, char *origem)
```

Essa função copia a string `origem` em `destino`. Ela também supõe que o tamanho de `destino` é maior ou igual ao de `origem`. O valor retornado é `destino`. Escreva uma função `main` que use a função `strcpy`.

6. Implemente a seguinte função usando ponteiros:

```
int strcmp(char *str1, char *str2)
```

Essa função retorna 0 se `str1 == str2`, retorna `-1` se `str1 < str2` e retorna 1 de `str1 > str2`.

Escreva uma função `main` que use a função `strcmp`.

7. Escreva um programa que leia uma sequência de caracteres de tamanho indeterminado do teclado e os armazene em um **vetor alocado dinamicamente**. O usuário irá digitar uma sequência de caracteres, sem limite de quantidade. Os caracteres serão digitados um a um e, sendo que caso ele deseje encerrar a entrada de dados, ele deverá teclar ENTER (`'\n'`).

**Saída:** Ao final do programa, exiba o vetor lido.

Os dados devem ser armazenados na memória deste modo:

- Inicie com um vetor de tamanho 10 alocado dinamicamente;
- Após, caso o vetor alocado esteja cheio, aloque um novo vetor do tamanho do vetor anterior adicionado espaço para mais 10 valores (tamanho `N+10`, onde `N` inicia com 10);
- Copie os valores já digitados da área inicial para esta área maior e libere a memória da área inicial;
- Repita este procedimento de expandir dinamicamente com mais 10 valores. Um novo vetor deve ser alocado cada vez que o atual estiver cheio. Assim o vetor irá ser 'expandido' de 10 em 10 valores.
- **Observação:** Use as funções `new` e `delete`. A fim de ler caracteres isolados do teclado, use a função `std::getchar()` ao invés de `cin >>`. A função `std::getchar()` retorna o caractere digitado no teclado.