

# Alocação Dinâmica

---

## Exercício 01

Escreva uma função que inverta o conteúdo de uma string. Por exemplo: UTFPR => RPFTU **Não é permitido utilizar funções da biblioteca string.h.**

Protótipo da função

```
void inverte_string(char* str);
```

Exemplo de utilização

```
char nome[10] = "Joao";  
inverte_string(nome);  
printf("%s", nome); //oaoJ
```

---

## Exercício 02

Escreva uma função que devolva a quantidade de vogais em uma string.

Desenvolva duas versões desta função:

- a) usando como saída o `return`
- b) usando como saída a estratégia do `scanf`.

**Não é permitido utilizar funções da biblioteca string.h.**

Protótipo da função

```
int qtde_vogais1(char* str);  
int qtde_vogais2(char* str, int* saida);
```

Exemplo de utilização

```
char texto[100] = "UTFPR Campo Mourao";  
int n1 = qtde_vogais1(texto);  
int n2;  
qtde_vogais2(texto, &n2);
```

## Exercício 03

Escreva uma função para criar uma cópia de uma string. **Não é permitido utilizar funções da biblioteca `string.h`.**

Protótipo da função

```
char* copia_string(char* str);
```

Exemplo de utilização

```
char texto[100] = "UTFPR Campo Mourao";  
char* str = copia_string(texto) ;
```

---

## Exercício 04

Escreva uma função para criar um vetor dinamicamente com um tamanho especificado e preenche-lo com um determinado valor. Desenvolva duas versões desta função:

- a) usando como saída o `return`
- b) usando como saída a estratégia do `scanf`.

Protótipo da função

```
int* cria_vetor1(int tam, int valor);  
bool cria_vetor2(int tam, int valor, int** saida);
```

Exemplo de utilização

```
int *v1, *v2;  
v1 = cria_vetor1(10, -1);  
cria_vetor2(5, 0, &v2);
```

---

## Exercício 05

Escreva uma função que crie um clone de um determinado vetor. Desenvolva duas versões desta função:

- a) usando como saída o `return`
- b) usando como saída a estratégia do `scanf`.

Obs.: Use os desenhos para auxiliar no raciocínio.

### Protótipo da função

```
int* clone1(int tam, int valor);  
bool clone2(int tam, int valor, int** saida);
```

### Exemplo de utilização

```
int v[5] = {2,4,6,8,10};  
int *copia1, *copia2;  
  
copia1 = vetor_cloneA(v, 5);  
vetor_cloneB(v, 5, &copia2);
```

---

## Exercício 06

Escreva uma função que crie um vetor preenchido com valores aleatórios.

### Protótipo da função

```
int* aleatorio1(int tam, int valor);  
bool aleatorio2(int tam, int valor, int** saida);
```

### Exemplo de utilização

```
int *v1 = aleatorio1(10);  
aleatorio2(100, &v2);
```

---

## Exercício 07

Escreva uma função que dado um vetor, dobre seu tamanho e devolva o valor do novo tamanho. A função deve preservar os valores do vetor e preencher com zeros as posições adicionais.

A função deve seguir os seguintes passos:

1. Alocar um novo vetor com o dobro do tamanho do vetor recebido por parâmetro.
2. Copiar os valores para o vetor novo
3. Preencher com 0 as novas posições

4. Desalocar o vetor antigo
5. Atualizar a referência do vetor recebido por parâmetro

Obs.: A função somente funciona para vetores alocados dinamicamente.

### Protótipo da função

```
int dobra_tamanho(int** v, int tam);
```

### Exemplo de utilização

```
int* v = (int*) calloc(3, sizeof(int));  
v[0] = 2;  
v[1] = 4;  
v[2] = 6;  
  
int novoTamanho = dobra_tamanho(&v, 3);  
//Resultado esperado  
// [2,4,6,0,0,0]
```

---

## Exercício 07

Escreva uma função para criar uma matriz dinamicamente com um tamanho especificado e preenche-la com um determinado valor.

### Protótipo da função

```
int** cria_matriz(int linhas, int colunas, int valor);
```

### Exemplo de utilização

```
int** m = cria_matriz(2, 3, -1);
```

---