

Exercícios sobre Alocação Dinâmica

Todos os exercícios devem possuir as seguintes características:

1. Modelo ENTRADA-PROCESSAMENTO-SAÍDA discutido em sala de aula.
2. Fornecer diferentes casos de teste para as funções.
3. Todas as funções deverão manipular valores do tipo `int`.

Exemplo

Escreva uma função para inicializar um vetor de valores inteiros com um determinado valor. A inicialização consiste em atribuir um mesmo valor a todas as posições do vetor.

Solução:

```
#include<stdio.h>

void vetor_inicializar(int* v, int tam, int valor){
    int i;
    for(i = 0; i < tam; i++){
        v[i] = valor;
        // *(v + i) = valor;
    }
}

int void main(){
    int v1[4];
    int v2[7];

    vetor_inicializar(v1, 4, -1);
    vetor_inicializar(v2, 7, 99);
    vetor_inicializar(v1, 4, 0);
}
```

Exercício 01

Escreva uma função para imprimir um vetor. A impressão deve ter o seguinte formato: `[1,2,3,4,5]`

Exercício 02

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

- a) usando a estratégia convencional
- b) usando a estratégia do `scanf`.

```
int main(){
    int *v1, *v2;
    v1 = vetor_novoA(10, -1);
    vetor_novoB(5, 0, &v2);
}
```

Exercício 03

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

- a) usando a estratégia convencional
- b) usando a estratégia do `scanf`.

Obs.: Use os desenhos para auxiliar no raciocínio.

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

    copia1 = vetor_cloneA(v, 5);
    vetor_cloneB(v, 5, &copia2);
}
```

Exercício 04

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

```
int main(){
    int *v1 = vetor_aleatorio(10);
    int *v2 = vetor_aleatorio(100);
}
```

Exercício 05

Escreva uma função que dado um vetor, adicione 1 ao valor de cada elemento.

```
int main(){
    int v1[5] = {10,20,30,40,50};

    vetor_incrementa(v1, 5); // [11,21,31,41,51]
}
```

Exercício 06

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.

```
int main(){
    int* v = (int*) calloc(3, sizeof(int));
    v[0] = 2;
    v[1] = 4;
    v[2] = 6;

    int novoTamanho = vetor_dobraTamanho(&v, 3);
    //Resultado esperado
    // [2,4,6,0,0,0]
}
```

Exercício 07

Considere o registro definido abaixo

```
typedef struct produto{
    unsigned int codigo;
    char[50] nome;
    float preco;
}
```

- a) Crie uma função responsável por criar dinamicamente um produto e inicializar seus valores. Os valores devem ser recebidos como parâmetro.
- b) Crie uma função responsável por imprimir um produto. A função deve percorrer e imprimir os valores de todos os atributos do produto.
- c) Escreva exemplos de utilização das funções criadas nos itens a e b.

Exercício 08

Considere o registro definido abaixo e o seguinte requisito:

- O registro para representar um aluno deve conter um vetor de float para armazenar suas notas. No entanto, a quantidade de notas dos alunos podem ser diferentes. Por exemplo, o aluno A pode ter 3 notas enquanto o aluno B pode ter 5 notas.

```
typedef struct aluno{  
    unsigned int codigo;  
    char[50] nome;  
    ...  
}
```

- Finalize o preenchimento da `struct` para contemplar satisfatoriamente o requisito descrito.
- Crie uma função para criar um aluno. Reflita sobre as informações necessárias para sua criação. Perceba que a função somente cria a estrutura, mas não inicializa seus valores como no exercício anterior.
- Escreva uma função para imprimir um aluno.
- Escreva exemplos de utilização das funções criadas nos itens `a`, `b` e `c`.