

Programação de Computadores II - BCC702

Lista de Exercícios

Ponteiros e Alocação Dinâmica de Memória

Exercício 01

Faça um programa que leia o tamanho de um vetor e seus elementos. Em seguida guarde em variáveis:

1. o valor do primeiro elemento do vetor;
2. o endereço da variável tamanho;
3. o endereço do vetor;
4. o endereço do primeiro elemento de do vetor;
5. o endereço do último elemento do vetor.

Por fim, escreva o valor de cada uma dessas variáveis. Normalmente os endereços serão diferentes para cada execução.

Informe o tamanho do vetor: 3

Informe os elementos do vetor: 5 1 8

Valor do primeiro elemento: 5

Endereço do tamanho: 0x7fff18f7a41c

Endereço do vetor: 0x7fff18f7a3f0

Endereço do primeiro elemento: 0x7fff18f7a3f0

Endereço do último elemento: 0x7fff18f7a3f8

Exercício 02

Para cada opção abaixo, qual é o valor variável *r*? Responda sem usar o computador.

a)

```
int x = 5, y = 2;
int *z = &x;
int *w = new int;
*w = *z;
int r = x + y + *w + *z;
```

b)

```
bool b1 = true, b2 = false, b3 = true;
bool *c = &b2;
bool d = b3;
bool r = b1 && *c && d;
```

c)

```
float a = 2.3, *b = &a, c = 4.5;
float v[3] = { 1.1, 2.2, 3.3 };
float *x = new float[3];
float *y = v;
x[0] = a;
```

```
x[1] = *b;
x[2] = y[2] + c;
float r = x[0] + x[1] + x[2];
```

Exercício 03

Considere o código fonte abaixo e responda o que aconteceria, em termos de gerenciamento de memória do computador, se as funções *aloc_estatica* e *aloc_dinamica* fossem usadas (uma de cada vez) no trecho destacado no código?

```
void aloca_estatica() {
    int i = 2;
}
void aloca_dinamica() {
    int *ptr = new int;
    *ptr = 2;
}
int main() {
    for(int i = 0; i < 10000000; i++) {
        #####
    }
    return 0;
}
```

Exercício 04

Faça um programa com um registro *Ponto*, que contém os campos numéricos *x* e *y*. Faça uma função que recebe por parâmetro dois pontos, calcula e retorna a distância euclidiana entre eles ($\text{sqrt}(x)$ = retorna a raiz quadrada de *x*).

Na função *main*, crie um vetor de pontos alocado dinamicamente de tamanho *N* (*N* informado pelo usuário) e leia as coordenadas de cada ponto. Use a função definida para calcular o tamanho do caminho definido pelos pontos, na ordem em que foram lidos.

```
Informe o número de pontos: 4
Informe as coordenadas: 3 3
Informe as coordenadas: 1 1
Informe as coordenadas: -1 2
Informe as coordenadas: 0 3
```

```
Tamanho do caminho: 6.47871
```