

## Algoritmos e Estrutura de Dados II

### Lista 2

**Aluno:** Leonardo Aguilar Murça

1. Um Ponteiro é:
  - a) O endereço de uma variável.
  - b) Uma variável que armazena endereços.**
  - c) O valor de uma variável.
  - d) Um indicador da próxima variável a ser acessada.
2. Escreva uma instrução em C++ que imprima o endereço da variável **var** do tipo int.

```
using namespace std;  
  
int main() {  
  
    int var;  
    cout << &var << endl;  
  
    return 0;  
}
```

3. Indique: (1) operador de endereços (2) operador de referência.
  - a) `p = &i;` (1)
  - b) `int &i = j;` (2)
  - c) `cout << &i;` (1)
  - d) `int *p = &i;` (1)
  - e) `int& func(void);` (2)
  - f) `void func(int &i);` (2)
  - g) `func (&i);` (1)
4. A instrução: `int *p;`
  - a) Cria um ponteiro com um valor indefinido.
  - b) Cria um ponteiro do tipo int.
  - c) Cria um ponteiro com o valor zero.
  - d) Cria um ponteiro que aponta para uma variável do tipo int.**

5. O que é do tipo `int` na instrução a seguir?

`int *p;`

- a) A variável `p`.
- b) O endereço de `p`.
- c) **A variável apontada por `p`.**
- d) O endereço da variável apontada por `p`.

6. Se o endereço de `var` for atribuído à um ponteiro variável `pvar`, quais das seguintes expressões são verdadeiras?

- a) `var == &pvar`; (**Falso**)
- b) `var == *pvar`; (**Verdadeiro**)
- c) `pvar = *var`; (**Falso**)
- d) `pvar == &var`; (**Verdadeiro**)

7. Qual a saída deste programa?

```
#include <iostream>

using namespace std;

int main() {
    int i=5, *p;
    p=&i;
    cout << p << "\\t" << (*p+2) << "\\t" << **&p
        << "\\t" << (3**p) << "t" << (**&p+4) << endl;
    return 0;
}
```

Saídas: 0x6dfeec(endereço de mem. de i)      7      5      15      9