Gabriel Martins de Almeida – RA 31010006371

Exercícios do Modulo 3

Slide 15 ----------

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>

using namespace std;

struct pessoas {

        char nom[30];
        char end[50];
        char cpf[11];
        char id[3];
};
int main(int argc, char * * argv) {
        pessoas nom[4];
        pessoas end[4];
        pessoas cpf[4];
        pessoas id[4];
        for (int i = 0; i < 4; i++) {
                cout << "Digite o nome da " << i+1 << " pessoa" << endl;
                gets(nomes[i].nom);
                cout << "Digite o endereo da " << i+1 << " pessoa" << endl;
                gets(end[i].end);
                cout << "Digite o cpf da " << i+1 << " pessoa" << endl;
                gets(cpf[i].cpf);
                cout << "Digite a idade da " << i+1 << " pessoa" << endl;
```

```
            gets(id[i].id);
        }
        for (int i = 0; i < 4; i++)  {
            printf("NOME:");
            printf(nom[i].nom);
            printf("ENDERECO:");
            printf(end[i].end);
            printf("CPF:");
            printf(cpf[i].cpf);
            printf("IDADE:");
            printf(id[i].id);
        }
        return 0;
}
```

Slide 38, 39, 40 e 41 ----------

Exercício 1 >

a) (V) O operador & permite-nos obter o endereço de uma variável. Permite também obter o endereço de um ponteiro.

b) (V) Se x é um inteiro e ptr um ponteiro para inteiros e ambos contêm no seu interior o número 100, então x+1 e ptr+1 apresentarão o número 101.

c) (F) O operador * nos permite obter o endereço de uma variável.

d) (V) Os ponteiros são variáveis que apontam para endereços na memória.

Exercício 2 > R: 5 7 5

Exercício 3 > R: 5 7 7

Exercício 4 > R: 5 20 20

Exercício 5 > R: *, antes do nome da variável que será o ponteiro.

Exercício 6> R: Possui o endereço da memória da variável.

Exercício 7 >

```
#include <iostream>


using namespace std;
```

```
int main(int argc, char * * argv) {
	int n[10] = {1,2,3,4,5,6,7,8,9,10};
	cout <<"impressão normal "<< endl;
	for (int i=0; i < 10; i++){
		int* pn = &n[i];
		cout<< *pn << ", " ;
	}
	cout << endl;
	cout <<"impressão inversa "<< endl;
	for ( int i = 9; i >= 0; i--) {
		int* pn = &n[i];
		cout<< *pn << ", " ;
}
	return 0;
}
```