

Lista 3

Entrega: 17/09/19

Lista

http://www.johnlenongardenghi.com.br/courses/2019_2/eda1/lista_ponteiros.pdf

Nota de Aula

http://www.johnlenongardenghi.com.br/courses/2019_2/eda1/aula_ponteiros.pdf

Usado para estudo

<http://linguagemc.com.br/ponteiros-em-c/>

<https://www.codingame.com/playgrounds/24988/programacao-c/operacoes-com-ponteiros>

<https://www.ime.usp.br/~pf/algoritmos/aulas/pont.html>

<http://www.dca.fee.unicamp.br/cursos/EA876/apostila/HTML/node142.html>

Questão 1

```
//v tem endereço inicial em 1000
int v[5] = {1, 2, 3, 4, 5};
int *ptr;
ptr = v;
```

ptr+1:

- 1004, porque é um deslocamento e a soma é feita em relação ao endereço e um inteiro ocupa 4 bytes

(*ptr)+1:

- 2, porque a soma é feita ao primeiro número do vetor, mas não é um deslocamento

*(ptr+1):

- 2, porque é um deslocamento e a soma é feita em relação ao vetor e um ponteiro aponta para o primeiro número de um vetor

*(ptr+10):

-

código usado para teste:

```
#include <stdio.h>

int main(){
    int i, offset;
    int b[] = { 10, 20, 30, 40, 50 };
    int *bPtr;
    bPtr = b;

    printf("%d\n", *bPtr);
    printf( "(bPtr) + 1 = %d\n", *(bPtr) + 1 );
    printf( "(bPtr + 1) = %d\n", *(bPtr + 1) );
    printf( "(bPtr + 10) = %d\n", *(bPtr + 10) );

    return 0;
}
```

Questão 4

```
int vetor[10];  
int *ponteiro;
```

vetor = vetor + 2:

- inválido

vetor++:

- inválido

vetor = ponteiro:

- inválido

ponteiro = vetor:

- válido

ponteiro = vetor + 2:

- inválido

Questão 5

```
void imprime (char *v, int n){  
    char *c;  
    for (c = v; c < v + n; c++)  
        printf ("%c", *c);  
}
```

- a função irá imprimir n caracteres a partir do endereço apontado por *v
1. a função recebe uma string
 2. o laço começa apontando v e c para o mesmo endereço
 3. quando c for igual ao endereço v+n, para

fonte: <https://pt.stackoverflow.com/questions/155557/d%C3%BAvida-em-rela%C3%A7%C3%A3o-a-ponteiros-em-c>

Questão 13

código usado para teste:

```
#include <stdio.h>  
  
void troca_int (int *x, int *y) {  
    int tmp;  
    tmp = *x;  
    *x = *y;  
    *y = tmp;  
}  
  
void troca_str (char *x, char *y) {  
    char tmp;  
    tmp = *x;  
    *x = *y;  
    *y = tmp;  
}  
  
int main() {  
    int a, b;  
    char *s1, *s2;
```

```
a = 456;
b = 123;

troca_int (&a, &b);

printf("a is %d\n", a);
printf("b is %d\n", b);

s1 = "Eu deveria aparecer depois";
s2 = "Eu deveria aparecer primeiro";

troca_str (&s1, &s2);

printf("s1 is %s\n", s1);
printf("s2 is %s\n", s2);
return 0;
}
```

Questão 15