

Estrutura de Dados e Algoritmos

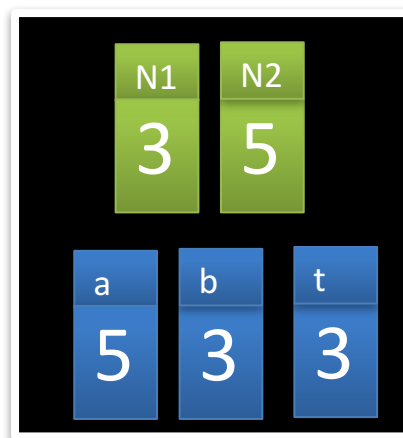
Uso de Ponteiros (Parâmetros por Referência)

- Procedimentos e Funções: Subconjuntos de códigos (comandos);
- Diferenças:
 - Procedimentos: não retorna nenhum valor. Consequentemente, não tem tipo;
 - Funções: retornam algum tipo de valor. Para tanto, devem ter o tipo do valor que retorna;
- Ambos podem ou não ter parâmetros;

- Exemplos:
 - `void ImprimeOla();`
 - `void ImprimeQtdeXPreco(int qtde, float preco);`
 - `float QuadradoDePI();`
 - `Long int SomaNumeros(int *VetNum, int Tamanho);`

- Passagem de parâmetros por valor:
 - A função recebe uma cópia da variável que é fornecida quando é invocada. Todas as alterações feitas dentro da função não vão afetar as variáveis globais:

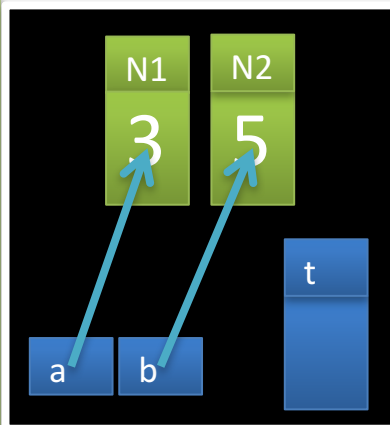
```
#include<stdio.h>
main()
{
    int N1=3, N2=5;
    troca(N1,N2);
}
```



```
void troca(int a, int b)
{
    int t;
    t=a;
    a=b;
    b=t;
}
```

- Passagem de parâmetros por referência:
 - A função recebe uma referência da variável que é fornecida quando é invocada. Todas as alterações feitas dentro da função vão afetar os variáveis globais:

```
#include<stdio.h>
main()
{
    int N1=3, N2=5;
    troca(&N1,&N2);
}
```

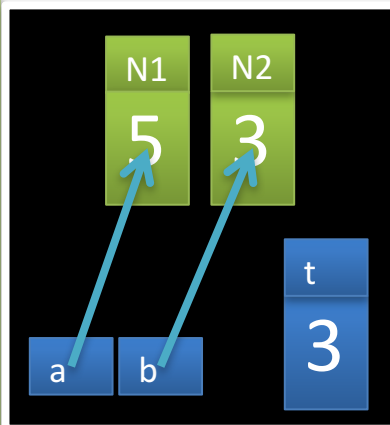


```
void troca(int *a, int *b)
{
    int t;
    t=*a;
    *a=*b;
    *b=t;
}
```

- Implementação em C:
 - Parâmetros por valor: OK;
 - Parâmetros por referência: NÃO;
- Implementação em C++:
 - Parâmetros por valor: OK;
 - Parâmetros por referência: OK;
- Alternativa para se implementar parâmetros por referência em linguagem C: Usar ponteiros.

- Passagem de parâmetros por referência:
 - A função recebe uma referência da variável que é fornecida quando é invocada. Todas as alterações feitas dentro da função vão afetar os variáveis globais:

```
#include<stdio.h>
main()
{
    int N1=3, N2=5;
    troca(&N1,&N2);
}
```

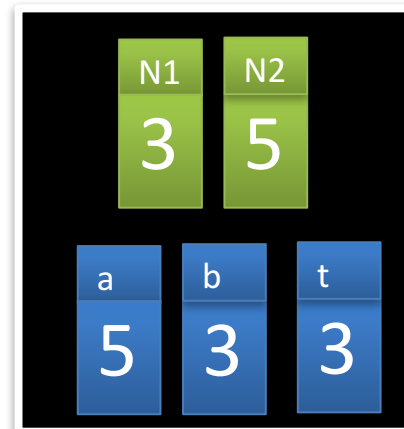


```
void troca(int *a, int *b)
{
    int t;
    t=*a;
    *a=*b;
    *b=t;
}
```

- Passagem de parâmetros por valor em C/C++:

```
#include<stdio.h>
main()
{
    int N1=3, N2=5;
    printf("Antes: %d, %d\n",N1,N2);
    troca(N1,N2);
    printf("Depois: %d, %d\n",N1,N2);
}
```

```
void troca(int a, int b)
{
    int t;
    t=a;
    a=b;
    b=t;
}
```

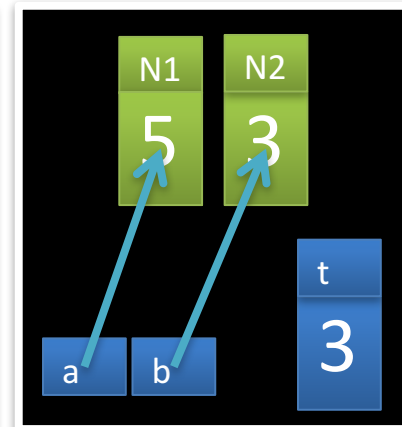


Antes: 3, 5
Depois: 3,5

- Passagem de parâmetros por referência em C:

```
#include<stdio.h>
main()
{
    int N1=3, N2=5;
    printf("Antes: %d, %d\n",N1,N2);
    troca(&N1, &N2);
    printf("Depois: %d, %d\n",N1,N2);
}
```

```
void troca(int *a, int *b)
{
    int t;
    t=*a;
    *a=*b;
    *b=t;
}
```

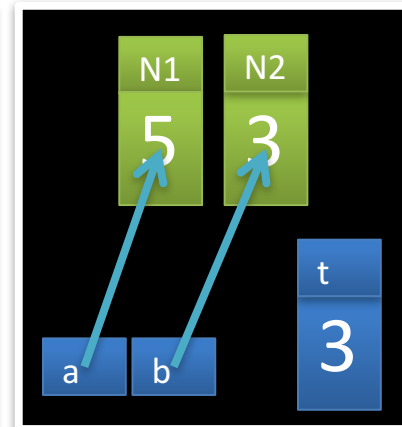


Antes: 3, 5
Depois: 5,3

- Passagem de parâmetros por referência em C++:

```
#include<stdio.h>
main()
{
    int N1=3, N2=5;
    printf("Antes: %d, %d\n",N1,N2);
    troca(N1, N2);
    printf("Depois: %d, %d\n",N1,N2);
}
```

```
void troca(int &a, int &b)
{
    int t;
    t=a;
    a=b;
    b=t;
}
```



Antes: 3, 5
Depois: 5,3

```
#include<stdio.h>
main()
{
    int N1=3, N2=5, N3=1;
    printf("Antes: %d, %d\n",N1,N2,N3);
    ordena(&N1,&N2,&N3);
    printf("Depois: %d, %d\n",N1,N2,N3);
}
```

Escreva a função ordena em Linguagem C de forma a fazer com que o programa principal imprima as mensagens abaixo:

Antes: 3, 5, 1
Depois: 1, 3, 5

```
#include<stdio.h>
main()
{
    int N1=3, N2=5, N3=1;
    printf("Antes: %d, %d\n",N1,N2,N3);
    ordena(N1,N2,N3);
    printf("Depois: %d, %d\n",N1,N2,N3);
}
```

Escreva a função ordena em Linguagem C++ de forma a fazer com que o programa principal imprima as mensagens abaixo:

Antes: 3, 5, 1
Depois: 1, 3, 5