



Passagem de parâmetros por referência

Ponteiros

- **Ponteiros** são variáveis utilizadas para guardar endereços de memória. Permitem acesso quase que irrestrito a toda a memória do sistema.
- Exemplo:

```
int *Odin;  
float *Loki;
```

Passagem de parâmetros por referência

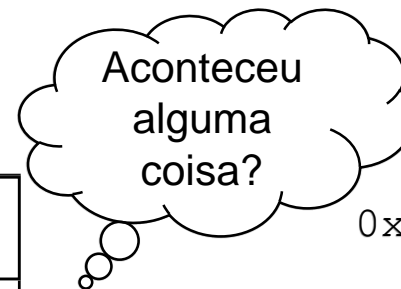
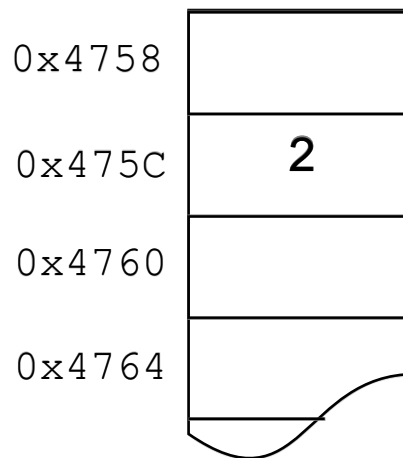
```
...  
/* Sub-rotinas */  
...
```

```
int main(void) {  
  
    int A;  
  
    scanf("%d", &A);  
  
    ProcEx1(A);  
  
}
```

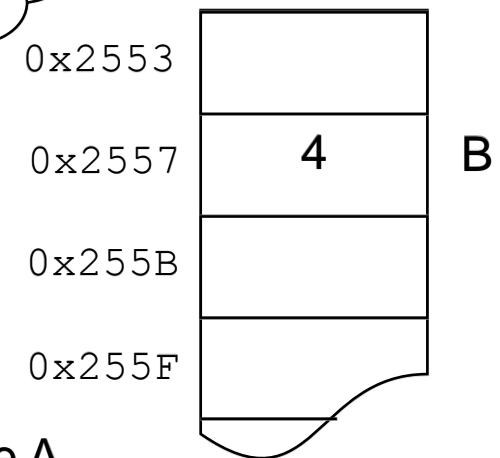
```
void ProcEx1(int B) {  
    B = 2*B;  
    printf("%d", B);  
}
```

4

Memória



Memória



Valor de A

Passagem de parâmetros por referência

...
/* Sub-rotinas */
...

```
int main(void) {
```

```
    int A;
```

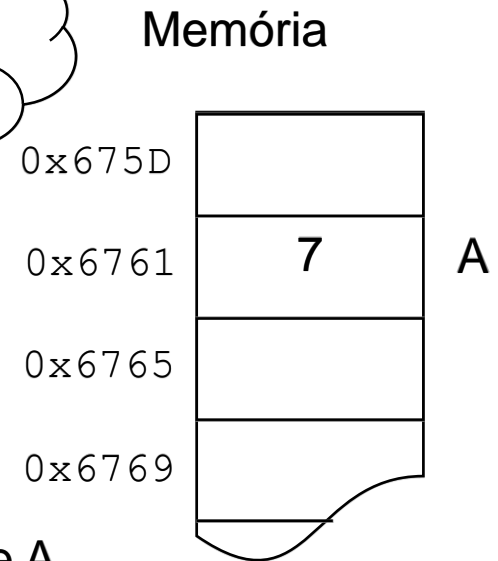
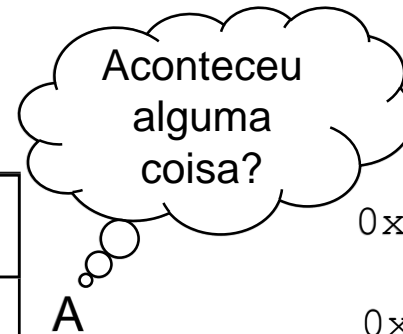
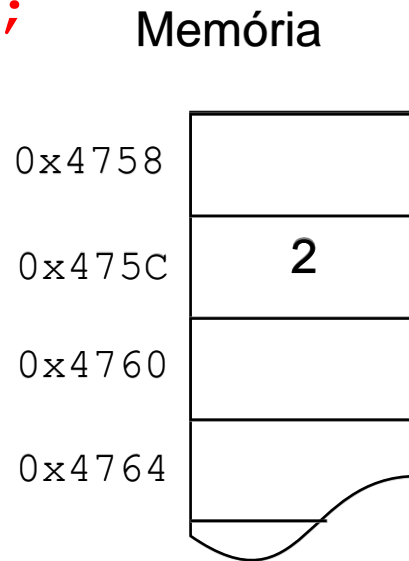
```
    scanf("%d", &A);
```

```
    ProcEx1(A);
```

```
    ProcEx2(A);
```

```
void ProcEx2(int A) {  
    A = 2*A+3;  
    printf("%d", A);  
}
```

7



Valor de A

Passagem de parâmetros por referência

...
/* Sub-rotinas */
...

```
int main(void) {
```

```
    int A;
```

```
    scanf("%d", &A);
```

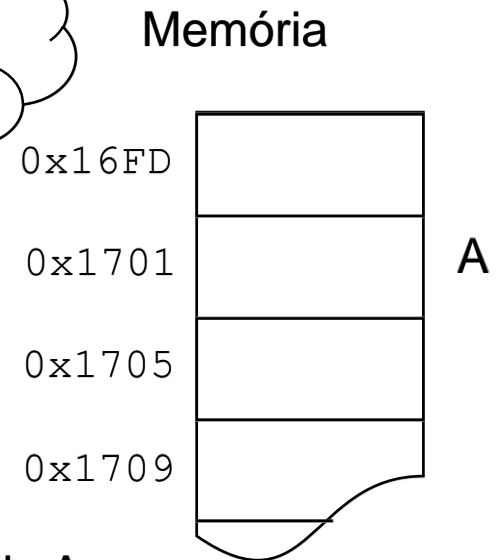
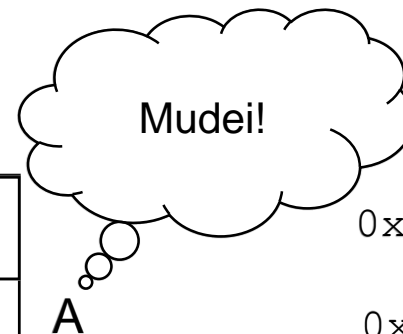
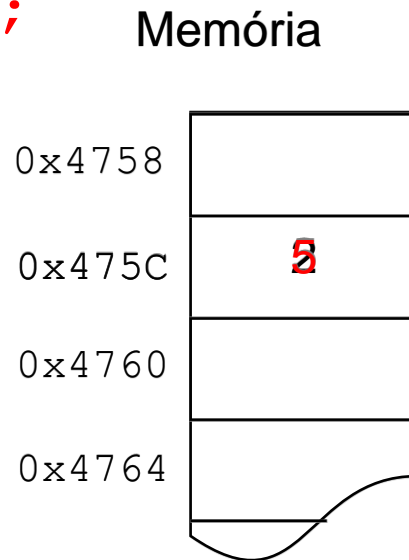
```
    ProcEx1(A);
```

```
    ProcEx2(A);
```

```
    ProcEx3(&A);
```

```
void ProcEx3(int *A) {  
    *A = 3 * (*A) - 1;  
    printf("%d", *A);  
}
```

5



Endereço de A

Passagem de parâmetros por referência

....
/* Sub-rotinas */
....

```
int main(void) {
```

```
    int A;
```

```
    scanf("%d", &A);
```

```
    ProcEx1(A);
```

```
    ProcEx2(A);
```

```
    ProcEx3(&A);
```

```
    ProcEx4(&A);
```

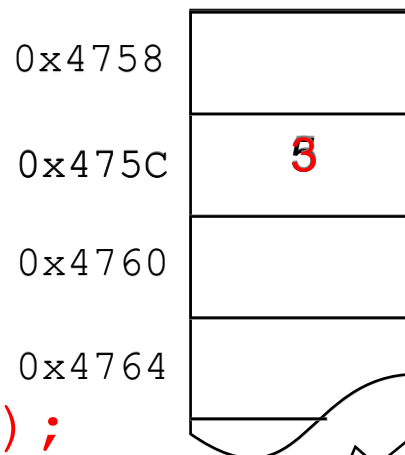
```
    printf("%d\n", A);
```

```
    return 0; }
```

```
void ProcEx4(int *B) {  
    *B = 2 + (*B) / 5;  
    printf("%d", *B);  
}
```

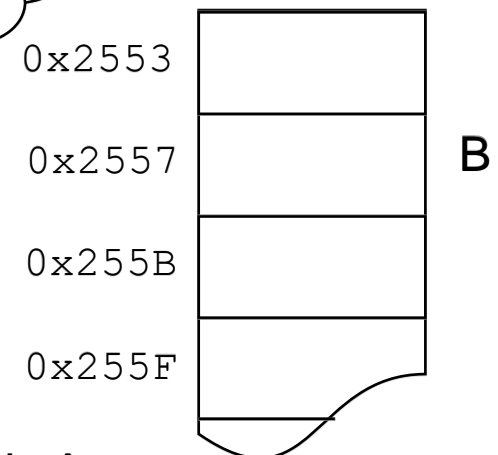
3

Memória



Mudei!

Memória



Endereço de A

3

- **Ponteiros** os ponteiros devem receber o endereço das variáveis manipuladas. *LEMBRAR É VIVER:* operador endereço - &.
- Exemplo:

```
int *Peter, Aranha;  
Peter = &Aranha;
```

- Exemplo:

```
int main()  
{  
    int IronMan, Thor, Capitao;  
    int *Shield;  
    Shield = &IronMan;  
    printf("Endereco: %p\n", &Shield);  
    printf("Endereco Apontado: %p\n", Shield);  
    *Shield = 69;  
    printf("Valor: %i\n", IronMan);  
    return 0;  
}
```


- **Ponteiros** podem ser utilizados como parâmetros de funções, endereçar vetores e estruturas.
- **Vetores** e **Ponteiros** realizam a mesma função, armazenam endereços.

- O acesso dos elementos de um vetor pode ser realizado utilizando ***notação de Ponteiros***
- Exemplo:

```
int *Atual;  
int Vetor[7], i;  
Atual = Vetor;  
for (i = 0; i < 7; i++)  
    *(Atual+i) = i;  
for (i = 0; i < 7; i++)  
    printf("%i\n", Vetor[i]);
```

Passagem de Parâmetros com Ponteiros

- Exemplo:

```
#define MAX 80
typedef struct{
    char nome[MAX];
    int QuantidadeFilmes;
} Vingador;

int main()
{
    Vingador HomemDeFerro;
    LerDados (&HomemDeFerro);
    printf("Nome Vingador: %s\nFilmes:%i\n",
        HomemDeFerro.nome,
        HomemDeFerro.QuantidadeFilmes);
    return 0;
}
```

Passagem de Parâmetros com Ponteiros

• Continuação Exemplo:

```
void LerDados (Vingador *A)
{
    printf("Informe o nome:");
    LerString( (*A).nome );
    printf("Quantidade de filmes:");
    scanf("%i", &(*A).QuantidadeFilmes );
}
```