

Programação 2

Ponteiros

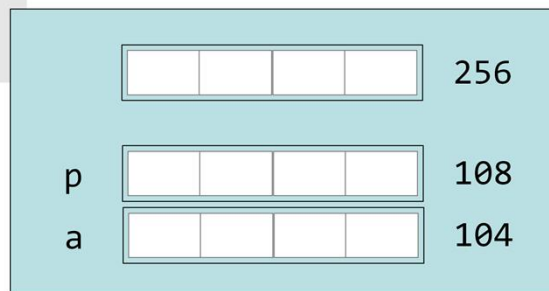
Rivera

Ponteiros

```
int main ( void )  
{  
    int a;  
    int *p;  
    p = &a;  
    *p = 2;  
    printf(" %d ", a);  
    return 0;  
}
```

Operadores unários

- & (“endereço de”)
✓ &a
- * (“conteúdo de”)
✓ *p



após a declaração da variável a e o vetor p, o vetor p passa a receber o endereço de a, e após isso, seu conteúdo passa a ser igual a 2, ou seja, o valor da variável a = 2.

Ponteiros: cuidados

```
int main ( void )
{
    int a, b, *p;
    a = 2;
    *p = 3;
    b = a + (*p);
    printf(" %d ", b);
    return 0;
}
```

- Erro em `*p = 3`
 - Memória apontada por `p` não inicializada
 - 3 armazenado num espaço de memória desconhecido

neste algoritmo, não houve a definição de alguma memória apontada por `p`, logo, não é possível alterar o valor de seu conteúdo, resultando em erro.

Funções que mudam valores de variáveis de outras

```
#include <stdio.h>
void troca(int a, int b);

int main (void)
{
    int a=10, b=20;
    troca(a,b);
    printf(" a=%d b= %d\n",a,b);
}

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

```
a = 10    b = 20
Press any key to continue
```

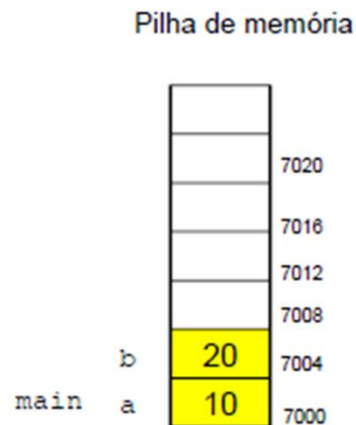
por meio do uso de uma variável auxiliar, os valores de a e b são trocados entre si e impressos na tela.

Funções que mudam valores de variáveis de outras

```
#include <stdio.h>
void troca(int a, int b);

int main (void)
{
    int a=10, b=20;
    → troca(a,b);
    printf(" a=%d b= %d\n",a,b);
}

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



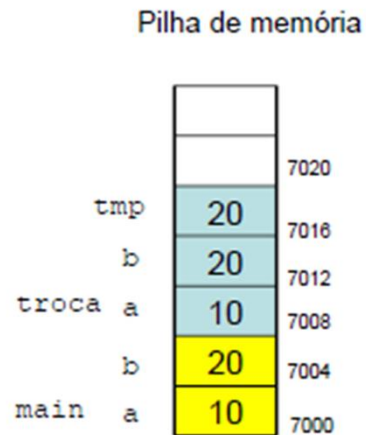
variáveis a e b inicializadas, com valores 10 e 20, respectivamente

Funções que mudam valores de variáveis de outras

```
#include <stdio.h>
void troca(int a, int b);

int main (void)
{
    int a=10, b=20;
    troca(a,b);
    printf(" a=%d b= %d\n",a,b);
}

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



a função troca é executada, criando-se uma variável auxiliar tmp que recebe o valor de B

Funções que mudam valores de variáveis de outras

```
#include <stdio.h>
void troca(int a, int b);

int main (void)
{
    int a=10, b=20;
    troca(a,b);
    printf(" a=%d b= %d\n",a,b);
}

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

Pilha de memória

		7020
tmp	20	7016
b	10	7012
troca a	10	7008
b	20	7004
main a	10	7000

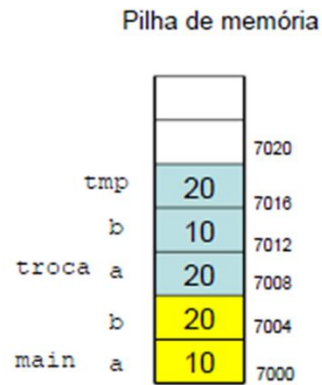
b recebe o valor de A

Funções que mudam valores de variáveis de outras

```
#include <stdio.h>
void troca(int a, int b);

int main (void)
{
    int a=10, b=20;
    troca(a,b);
    printf(" a=%d b= %d\n",a,b);
}

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



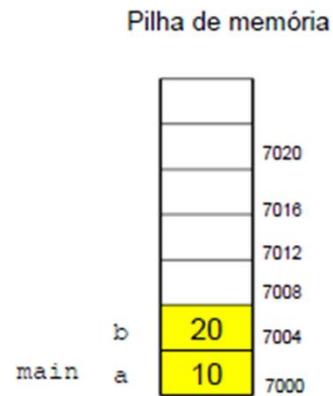
A recebe o valor da variavel tmp, anteriormente igual ao valor de B

Funções que mudam valores de variáveis de outras

```
#include <stdio.h>
void troca(int a, int b);

int main (void)
{
    int a=10, b=20;
    troca(a,b);
    printf(" a=%d b= %d\n",a,b);
}

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



as variáveis com os valores trocados são impressas, e depois limpas da memória, restando apenas as variáveis originais, com seus valores intactos, pois uma função não pode mudar o valor da variável de outra, apenas criar variáveis temporárias com os novos valores, o que foi o caso.

Ponteiros

- Passagem de ponteiros para funções:
 - ♦ Função g chama função f
 - f não pode alterar diretamente valores de variáveis de g
 - se g passar para f os valores dos endereços de memória onde as variáveis de g estão armazenadas, f pode alterar, indiretamente, os valores das variáveis de g

```
#include <stdio.h>
void troca(int *pa, int *pb);
int main (void)
{
    int a=10, b=20;
    troca(&a,&b);
    printf(" a = %d b = %d \n", a, b);
}
```

```
void troca(int *pa, int *pb)
{
    int tmp=*pb;
    *pb=*pa;
    *pa=tmp;
}
```

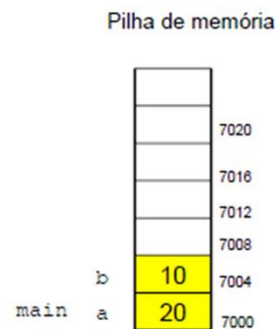
a=20 b=10
Press any key to continue

a função main passa para a função de troca os endereços de a e b, que nesta passam a serem apontados por pa e pb, que por sua vez, trocam seus conteúdos, alterando o valor de a e b, mesmo quando fora do escopo da função main.

Ponteiros

```
#include <stdio.h>
void troca(int *pa, int *pb);
int main (void)
{
    int a=10, b=20;
    → troca(&a,&b);
    printf(" a = %d b = %d \n", a, b);
}

void troca(int *pa, int *pb) {
    int tmp=*pb;
    *pb=*pa;
    *pa=tmp;
}
```



variáveis a e b inicializadas, com seus respectivos valores

Ponteiros

```
#include <stdio.h>
void troca(int *pa, int *pb);
int main (void)
{
    int a=10, b=20;
    troca(&a,&b);
    printf(" a = %d b = %d \n", a, b);
}
```

→

```
void troca(int *pa, int *pb) {
    int tmp=*pb;
    *pb=*pa;
    *pa=tmp;
}
```

Pilha de memória

		7020
tmp	-	7016
pb	7004	7012
troca pa	7000	7008
b	20	7004
main a	10	7000

função troca executada, inicializando pb e pa, apontando para os endereços de a e b, respectivamente

Ponteiros

```
#include <stdio.h>
void troca(int *pa, int *pb);
int main (void)
{
    int a=10, b=20;
    troca(&a,&b);
    printf(" a = %d b = %d \n", a, b);
}
```

```
void troca(int *pa, int *pb) {
    int tmp=*pb;
    *pb=*pa;
    *pa=tmp;
}
```



Pilha de memória

		7020
		7016
tmp	20	7012
pb	7004	7008
troca pa	7000	7004
b	20	7000
main a	10	

tmp inicializada, recebendo o valor do conteúdo de pb (b)

Ponteiros

```
#include <stdio.h>
void troca(int *pa, int *pb);
int main (void)
{
    int a=10, b=20;
    troca(&a,&b);
    printf(" a = %d b = %d \n", a, b);
}

void troca(int *pa, int *pb) {
    int tmp=*pb;
    *pb=*pa;
    *pa=tmp;
}
```



Pilha de memória

			7020
tmp	20		7016
pb	7004		7012
troca pa	7000		7008
b	10		7004
main a	10		7000

o conteúdo de pb (b) passa a valer o conteúdo de pa (a)

Ponteiros

```
#include <stdio.h>
void troca(int *pa, int *pb);
int main (void)
{
    int a=10, b=20;
    troca(&a,&b);
    printf(" a = %d b = %d \n", a, b);
}

void troca(int *pa, int *pb) {
    int tmp=*pb;
    *pb=*pa;
    *pa=tmp;
}
```



Pilha de memória

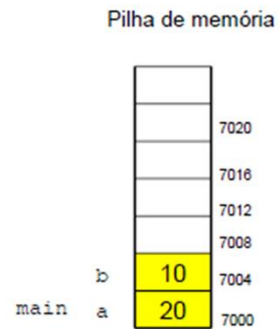
			7020
tmp	20		7016
pb	7004		7012
troca pa	7000		7008
b	10		7004
main a	20		7000

o conteúdo de pa (a) passa a valer tmp (20)

Ponteiros

```
#include <stdio.h>
void troca(int *pa, int *pb);
int main (void)
{
    int a=10, b=20;
    troca(&a,&b);
    printf(" a = %d b = %d \n", a, b);
}

void troca(int *pa, int *pb) {
    int tmp=*pb;
    *pb=*pa;
    *pa=tmp;
}
```



o valor atualizado de a e b é emitido na tela, e a pilha de execução é limpa, restando apenas a e b, com seus valores devidamente modificados nas variáveis originais, por meio de outra função, utilizando ponteiros.

Trabalho

- Implementar um programa que faça:
 - Defina um valor para n (número de elementos)
 - Digite os n elementos inteiros
 - Cada elemento digitado deve colocar na sequência certa (de menor a maior)
 - Mostrar o resultado ordenado