

# Unidade V: Árvore Binária - Inserção em C++ com passagem por referência




**PUC Minas**

Instituto de Ciências Exatas e Informática  
Departamento de Ciência da Computação

# Agenda

- Passagem de Parâmetros por Referência (C++)
- Estrutura dos Arquivos
- Classe Nó
- Classe ArvoreBinaria

# Agenda

- **Passagem de Parâmetros por Referência (C++)** 
- Estrutura dos Arquivos
- Classe Nó
- Classe ArvoreBinaria

# Passagem de Parâmetro

- As linguagens de programação normalmente permitem as passagens de parâmetro por valor e por referência
- A Linguagem C (como o Java) permite **somente** a passagem de parâmetro por valor
- Na passagem de parâmetros por valor, passamos **apenas** o valor e qualquer alteração no método chamado não será refletida no que chama

# Passagem de Parâmetro

- Um erro comum na linguagem C (no Java também) é achar que ela tem a passagem de parâmetros por referência e essa confusão acontece quando o argumento é um ponteiro

# Passagem de Parâmetros por Referência

- Passamos uma referência (**apelido**), fazendo com que qualquer alteração no método chamado seja refletida no que chama
- Nesse caso, o argumento do método chamado ocupa a mesma área de memória da variável correspondente no método que chama
- Por exemplo, as linguagens C++ e C# possuem a passagem de parâmetros por referência
- O uso desse tipo de passagem de parâmetros **tem que ser explicitado**

# Passagem de Parâmetros por Referência em C++

- Indicamos a passagem de parâmetros por referência inserindo um & na declaração da função, entre o tipo e o nome da variável
- No exemplo abaixo, a variável x é passada por referência e y, por valor

//declaração da função

```
void funcao(int& x, int y){ ... }
```

//chamada da função

```
...  
funcao(x, y);
```

# Passagem de Parâmetros por Referência em C#

- Indicamos a passagem de parâmetros por referência inserindo as palavras **out** e **ref** na declaração e na chamada da função. Na declaração, as palavras **out** e **ref** aparecem antes do tipo da variável
- No exemplo abaixo, as variáveis x1 e x2 são passadas por referência e y, por valor

//declaração da função

```
void funcao(out int x1, ref int x2, int y){ ... }
```

//chamada da função

```
...  
funcao(out x1, ref x2, y);
```



# Exemplo de Passagem por Valor com Ponteiro

```
void funcao(int* a, int b){  
    *a = *a + 1;  
    b = b + 1;  
    printf("\n(%p) (%i) (%i)", a, *a, b);  
}
```

```
int main(int argc, char *argv[]) {  
    int a = 0, b = 0;  
    funcao(&a, b);  
    printf("\n(%i) (%i)", a, b);  
    return 0;  
}
```

- Faça o quadro de memória e mostre a saída na tela do programa acima

# Exemplo de Passagem por Valor com Ponteiro

```
void funcao(int* a, int b){  
    *a = *a + 1;  
    b = b + 1;  
    printf("\n(%p) (%i) (%i)", a, *a, b);  
}
```

```
int main(int argc, char *argv[]) {  
    int a = 0, b = 0;  
    funcao(&a, b);  
    printf("\n(%i) (%i)", a, b);  
    return 0;  
}
```

**Tela**

**Memória**

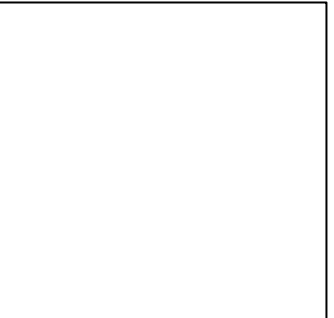
- Faça o quadro de memória e mostre a saída na tela do programa acima

# Exemplo de Passagem por Valor com Ponteiro

```
void funcao(int* a, int b){  
    *a = *a + 1;  
    b = b + 1;  
    printf("\n(%p) (%i) (%i)", a, *a, b);  
}
```

```
int main(int argc, char *argv[]) {  
    int a = 0, b = 0;  
    funcao(&a, b);  
    printf("\n(%i) (%i)", a, b);  
    return 0;  
}
```

**Tela**



**Memória**

a

0

33h

...

b

0

51h

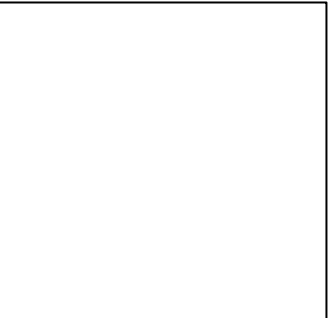
- Faça o quadro de memória e mostre a saída na tela do programa acima

# Exemplo de Passagem por Valor com Ponteiro

```
void funcao(int* a, int b){  
    *a = *a + 1;  
    b = b + 1;  
    printf("\n(%p) (%i) (%i)", a, *a, b);  
}
```

```
int main(int argc, char *argv[]) {  
    int a = 0, b = 0;  
    funcao(&a, b);  
    printf("\n(%i) (%i)", a, b);  
    return 0;  
}
```

**Tela**



**Memória**

a

0

33h

...

b

0

51h

- Faça o quadro de memória e mostre a saída na tela do programa acima

# Exemplo de Passagem por Valor com Ponteiro

```
void funcao(int* a, int b){
    *a = *a + 1;
    b = b + 1;
    printf("\n(%p) (%i) (%i)", a, *a, b);
}

int main(int argc, char *argv[]) {
    int a = 0, b = 0;
    funcao(&a, b);
    printf("\n(%i) (%i)", a, b);
    return 0;
}
```

**Tela**

**Memória**

a	0	33h
	...	
b	0	51h
	...	
a	33h	7Bh
	...	
b	0	C2h

- Faça o quadro de memória e mostre a saída na tela do programa acima

# Exemplo de Passagem por Valor com Ponteiro

```
void funcao(int* a, int b){
    *a = *a + 1;
    b = b + 1;
    printf("\n(%p) (%i) (%i)", a, *a, b);
}

int main(int argc, char *argv[]) {
    int a = 0, b = 0;
    funcao(&a, b);
    printf("\n(%i) (%i)", a, b);
    return 0;
}
```

**Tela**

**Memória**

a	1	33h
	...	
b	0	51h
	...	
a	33h	7Bh
	...	
b	0	C2h

- Faça o quadro de memória e mostre a saída na tela do programa acima

# Exemplo de Passagem por Valor com Ponteiro

```
void funcao(int* a, int b){
    *a = *a + 1;
    b = b + 1;
    printf("\n(%p) (%i) (%i)", a, *a, b);
}

int main(int argc, char *argv[]) {
    int a = 0, b = 0;
    funcao(&a, b);
    printf("\n(%i) (%i)", a, b);
    return 0;
}
```

**Tela**

**Memória**

a	1	33h
	...	
b	0	51h
	...	
a	33h	7Bh
	...	
b	1	C2h

- Faça o quadro de memória e mostre a saída na tela do programa acima

# Exemplo de Passagem por Valor com Ponteiro

```
void funcao(int* a, int b){  
    *a = *a + 1;  
    b = b + 1;  
    printf("\n(%p) (%i) (%i)", a, *a, b);  
}
```

```
int main(int argc, char *argv[]) {  
    int a = 0, b = 0;  
    funcao(&a, b);  
    printf("\n(%i) (%i)", a, b);  
    return 0;  
}
```

**Tela**

(33h) (1) (1)

**Memória**

a	1	33h
	...	
b	0	51h
	...	
a	33h	7Bh
	...	
b	1	C2h

- Faça o quadro de memória e mostre a saída na tela do programa acima



# Exemplo de Passagem por Valor com Ponteiro

```
void funcao(int* a, int b){
    *a = *a + 1;
    b = b + 1;
    printf("\n(%p) (%i) (%i)", a, *a, b);
}
```

```
int main(int argc, char *argv[]) {
    int a = 0, b = 0;
    funcao(&a, b);
    printf("\n(%i) (%i)", a, b);
    return 0;
}
```

**Tela**

(33h) (1) (1)

**Memória**

a	1	33h
	...	
b	0	51h

- Faça o quadro de memória e mostre a saída na tela do programa acima

# Exemplo de Passagem por Valor com Ponteiro

```
void funcao(int* a, int b){  
    *a = *a + 1;  
    b = b + 1;  
    printf("\n(%p) (%i) (%i)", a, *a, b);  
}
```

```
int main(int argc, char *argv[]) {  
    int a = 0, b = 0;  
    funcao(&a, b);  
    printf("\n(%i) (%i)", a, b);  
    return 0;  
}
```

**Tela**

(33h) (1) (1)  
(1)(0)

**Memória**

a	1	33h
	...	
b	0	51h

- Faça o quadro de memória e mostre a saída na tela do programa acima

# Exemplo de Passagem por Valor com Ponteiro

```
void funcao(int* a, int b){  
    *a = *a + 1;  
    b = b + 1;  
    printf("\n(%p) (%i) (%i)", a, *a, b);  
}
```

```
int main(int argc, char *argv[]) {  
    int a = 0, b = 0;  
    funcao(&a, b);  
    printf("\n(%i) (%i)", a, b);  
    return 0;  
}
```

**Tela**

(33h) (1) (1)  
(1)(0)

**Memória**

a	1	33h
	...	
b	0	51h

- Faça o quadro de memória e mostre a saída na tela do programa acima

# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){
    int z;
    printf(" %i %i ?", x, y);
    x = y = z = 2;
    printf("\n %i %i %d", x, y, z);
    return z;
}

int main(int argc, char **argv){
    int a, b, c;
    a = b = 1;
    printf(" %i %i ?", a, b);
    c = metodo(a, b);
    printf("\n %i %i %d", a, b, c);
}
```

**Tela**

**Memória**

- Faça o quadro de memória e mostre a saída na tela do programa

# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){  
    int z;  
    printf(" %i %i ?", x, y);  
    x = y = z = 2;  
    printf("\n %i %i %d", x, y, z);  
    return z;  
}
```

```
int main(int argc, char **argv){  
    int a, b, c;  
    a = b = 1;  
    printf(" %i %i ?", a, b);  
    c = metodo(a, b);  
    printf("\n %i %i %d", a, b, c);  
}
```

**Tela**

**Memória**

- Faça o quadro de memória e mostre a saída na tela do programa

# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){  
    int z;  
    printf(" %i %i ?", x, y);  
    x = y = z = 2;  
    printf("\n %i %i %d", x, y, z);  
    return z;  
}
```

```
int main(int argc, char **argv){  
    int a, b, c;  
    a = b = 1;  
    printf(" %i %i ?", a, b);  
    c = metodo(a, b);  
    printf("\n %i %i %d", a, b, c);  
}
```

**Tela**

**Memória**

a  
b  
c

?
?
?

- Faça o quadro de memória e mostre a saída na tela do programa

# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){  
    int z;  
    printf(" %i %i ?", x, y);  
    x = y = z = 2;  
    printf("\n %i %i %d", x, y, z);  
    return z;  
}
```

```
int main(int argc, char **argv){  
    int a, b, c;  
    a = b = 1;  
    printf(" %i %i ?", a, b);  
    c = metodo(a, b);  
    printf("\n %i %i %d", a, b, c);  
}
```

**Tela**

**Memória**

a  
b  
c

1
1
?

- Faça o quadro de memória e mostre a saída na tela do programa

# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){  
    int z;  
    printf(" %i %i ?", x, y);  
    x = y = z = 2;  
    printf("\n %i %i %d", x, y, z);  
    return z;  
}
```

```
int main(int argc, char **argv){  
    int a, b, c;  
    a = b = 1;  
    printf(" %i %i ?", a, b);  
    c = metodo(a, b);  
    printf("\n %i %i %d", a, b, c);  
}
```

**Tela**

1 1 ?

**Memória**

a  
b  
c

1

1

?

- Faça o quadro de memória e mostre a saída na tela do programa



# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){  
    int z;  
    printf(" %i %i ?", x, y);  
    x = y = z = 2;  
    printf("\n %i %i %d", x, y, z);  
    return z;  
}
```

```
int main(int argc, char **argv){  
    int a, b, c;  
    a = b = 1;  
    printf(" %i %i ?", a, b);  
    c = metodo(a, b);  
    printf("\n %i %i %d", a, b, c);  
}
```

**Tela**

1 1 ?

**Memória**

a  
b  
c

1

1

?

- Faça o quadro de memória e mostre a saída na tela do programa

# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){  
    int z;  
    printf(" %i %i ?", x, y);  
    x = y = z = 2;  
    printf("\n %i %i %d", x, y, z);  
    return z;  
}
```

```
int main(int argc, char **argv){  
    int a, b, c;  
    a = b = 1;  
    printf(" %i %i ?", a, b);  
    c = metodo(a, b);  
    printf("\n %i %i %d", a, b, c);  
}
```

**Tela**

1 1 ?

**Memória**

a/x

1

b

1

c

?

y

1

- Faça o quadro de memória e mostre a saída na tela do programa

# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){
```

```
    int z;
```

```
    printf(" %i %i ?", x, y);
```

```
    x = y = z = 2;
```

```
    printf("\n %i %i %d", x, y, z);
```

```
    return z;
```

```
}
```

```
int main(int argc, char **argv){
```

```
    int a, b, c;
```

```
    a = b = 1;
```

```
    printf(" %i %i ?", a, b);
```

```
    c = metodo(a, b);
```

```
    printf("\n %i %i %d", a, b, c);
```

```
}
```

Tela

1 1 2

- **x** é passado por referência, logo, **x** e **a** ocupam a mesma posição de memória
- **y** é passado por valor, logo, **y** e **b** têm posições de memória distintas

Memória

a/x

1

b

1

c

?

y

1

- Faça o quadro de memória e mostre a saída na tela do programa

# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){
    int z;
    printf(" %i %i ?", x, y);
    x = y = z = 2;
    printf("\n %i %i %d", x, y, z);
    return z;
}

int main(int argc, char **argv){
    int a, b, c;
    a = b = 1;
    printf(" %i %i ?", a, b);
    c = metodo(a, b);
    printf("\n %i %i %d", a, b, c);
}
```

**Tela**

1 1 ?

**Memória**

a/x

1

b

1

c

?

y

1

z

?

- Faça o quadro de memória e mostre a saída na tela do programa

# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){
    int z;
    printf(" %i %i ?", x, y);
    x = y = z = 2;
    printf("\n %i %i %d", x, y, z);
    return z;
}

int main(int argc, char **argv){
    int a, b, c;
    a = b = 1;
    printf(" %i %i ?", a, b);
    c = metodo(a, b);
    printf("\n %i %i %d", a, b, c);
}
```

## Tela

1	1	?
1	1	?

## Memória

a/x

1

b

1

c

?

y

1

z

?

- Faça o quadro de memória e mostre a saída na tela do programa

# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){
    int z;
    printf(" %i %i ?", x, y);
    x = y = z = 2;
    printf("\n %i %i %d", x, y, z);
    return z;
}

int main(int argc, char **argv){
    int a, b, c;
    a = b = 1;
    printf(" %i %i ?", a, b);
    c = metodo(a, b);
    printf("\n %i %i %d", a, b, c);
}
```

## Tela

1	1	?
1	1	?

## Memória

a/x

2

b

1

c

?

y

2

z

2

- Faça o quadro de memória e mostre a saída na tela do programa

# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){
    int z;
    printf(" %i %i ?", x, y);
    x = y = z = 2;
    printf("\n %i %i %d", x, y, z);
    return z;
}

int main(int argc, char **argv){
    int a, b, c;
    a = b = 1;
    printf(" %i %i ?", a, b);
    c = metodo(a, b);
    printf("\n %i %i %d", a, b, c);
}
```

## Tela

1	1	?
1	1	?
2	2	2

## Memória

a/x	2
b	1
c	?
y	2
z	2

- Faça o quadro de memória e mostre a saída na tela do programa

# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){  
    int z;  
    printf(" %i %i ?", x, y);  
    x = y = z = 2;  
    printf("\n %i %i %d", x, y, z);  
    return z;  
}
```

```
int main(int argc, char **argv){  
    int a, b, c;  
    a = b = 1;  
    printf(" %i %i ?", a, b);  
    c = metodo(a, b);  
    printf("\n %i %i %d", a, b, c);  
}
```

## Tela

1	1	?
1	1	?
2	2	2

## Memória

a/x	2
b	1
c	?
y	2
z	2

- Faça o quadro de memória e mostre a saída na tela do programa



# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){  
    int z;  
    printf(" %i %i ?", x, y);  
    x = y = z = 2;  
    printf("\n %i %i %d", x, y, z);  
    return z;  
}
```

```
int main(int argc, char **argv){  
    int a, b, c;  
    a = b = 1;  
    printf(" %i %i ?", a, b);  
    c = metodo(a, b);  
    printf("\n %i %i %d", a, b, c);  
}
```

## Tela

1	1	?
1	1	?
2	2	2

## Memória

a/x	2
b	1
c	2
y	2
z	2

- Faça o quadro de memória e mostre a saída na tela do programa

# Exemplo (1) de Passagem por Referência em C++

```
int metodo(int& x, int y){  
    int z;  
    printf(" %i %i ?", x, y);  
    x = y = z = 2;  
    printf("\n %i %i %d", x, y, z);  
    return z;  
}
```

```
int main(int argc, char **argv){  
    int a, b, c;  
    a = b = 1;  
    printf(" %i %i ?", a, b);  
    c = metodo(a, b);  
    printf("\n %i %i %d", a, b, c);  
}
```

## Tela

1	1	?
1	1	?
2	2	2
2	1	2

## Memória

a/x

2

b

1

c

2

y

2

z

2

- Faça o quadro de memória e mostre a saída na tela do programa

## Exemplo (2) de Passagem por Referência em C++

```
int* metodo(int& a, int* b){  
    a = 5;  
    *b = 6;  
    int* resp = new int[3];  
    resp[0] = 10; resp[1] = 20; resp[2] = 30;  
    return resp;  
}  
  
int main(int argc, char **argv){  
    int a = 10, b = 25;  
    int* vet = metodo(a, &b);  
    printf("%d %d", a, b);  
    printf("%d %d %d", vet[0], vet[1], vet[2]);  
}
```

- Faça o quadro de memória e mostre a saída na tela do programa

# Exemplo (2) de Passagem por Referência em C++

```
int* metodo(int& a, int* b){  
    a = 5;  
    *b = 6;  
    int* resp = new int[3];  
    resp[0] = 10; resp[1] = 20; resp[2] = 30;  
    return resp;  
}
```

```
int main(int argc, char **argv){  
    int a = 10, b = 25;  
    int* vet = metodo(a, &b);  
    printf("%d %d", a, b);  
    printf("%d %d %d", vet[0], vet[1], vet[2]);  
}
```

**Tela**

**Memória**

# Exemplo (2) de Passagem por Referência em C++

```
int* metodo(int& a, int* b){  
    a = 5;  
    *b = 6;  
    int* resp = new int[3];  
    resp[0] = 10; resp[1] = 20; resp[2] = 30;  
    return resp;  
}
```

```
int main(int argc, char **argv){  
    int a = 10, b = 25;  
    int* vet = metodo(a, &b);  
    printf("%d %d", a, b);  
    printf("%d %d %d", vet[0], vet[1], vet[2]);  
}
```

**Tela**

**Memória**

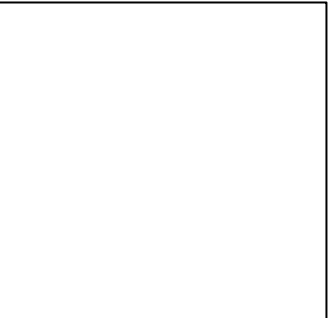
a	10	33h
	...	
b	25	51h

# Exemplo (2) de Passagem por Referência em C++

```
int* metodo(int& a, int* b){  
    a = 5;  
    *b = 6;  
    int* resp = new int[3];  
    resp[0] = 10; resp[1] = 20; resp[2] = 30;  
    return resp;  
}
```

```
int main(int argc, char **argv){  
    int a = 10, b = 25;  
    int* vet = metodo(a, &b);  
    printf("%d %d", a, b);  
    printf("%d %d %d", vet[0], vet[1], vet[2]);  
}
```

**Tela**



**Memória**

a	10	33h
	...	
b	25	51h

# Exemplo (2) de Passagem por Referência em C++

```
int* metodo(int& a, int* b){  
    a = 5;  
    *b = 6;  
    int* resp = new int[3];  
    resp[0] = 10; resp[1] = 20; resp[2] = 30;  
    return resp;  
}  
  
int main(int argc, char **argv){  
    int a = 10, b = 25;  
    int* vet = metodo(a, &b);  
    printf("%d %d", a, b);  
    printf("%d %d %d", vet[0], vet[1], vet[2]);  
}
```

**Tela**

**Memória**

a/a	10	33h
	...	
b	25	51h
	...	
b	51h	7Bh

# Exemplo (2) de Passagem por Referência em C++

```
int* metodo(int& a, int* b){  
    a = 5;  
    *b = 6;  
    int* resp = new int[3];  
    resp[0] = 10; resp[1] = 20; resp[2] = 30;  
    return resp;  
}  
  
int main(int argc, char **argv){  
    int a = 10, b = 25;  
    int* vet = metodo(a, &b);  
    printf("%d %d", a, b);  
    printf("%d %d %d", vet[0], vet[1], vet[2]);  
}
```

**Tela**

**Memória**

a/a	5	33h
	...	
b	25	51h
	...	
b	51h	7Bh



# Exemplo (2) de Passagem por Referência em C++

```
int* metodo(int& a, int* b){  
    a = 5;  
    *b = 6;  
  
    int* resp = new int[3];  
    resp[0] = 10; resp[1] = 20; resp[2] = 30;  
    return resp;  
}  
  
int main(int argc, char **argv){  
    int a = 10, b = 25;  
    int* vet = metodo(a, &b);  
    printf("%d %d", a, b);  
    printf("%d %d %d", vet[0], vet[1], vet[2]);  
}
```

**Tela**

**Memória**

a/a	5	33h
	...	
b	6	51h
	...	
b	51h	7Bh

# Exemplo (2) de Passagem por Referência em C++

```
int* metodo(int& a, int* b){  
    a = 5;  
    *b = 6;  
    int* resp = new int[3];  
    resp[0] = 10; resp[1] = 20; resp[2] = 30;  
    return resp;  
}  
  
int main(int argc, char **argv){  
    int a = 10, b = 25;  
    int* vet = metodo(a, &b);  
    printf("%d %d", a, b);  
    printf("%d %d %d", vet[0], vet[1], vet[2]);  
}
```

E9h

--	--	--

Tela

Memória

a/a	5	33h
	...	
b	6	51h
	...	
b	51h	7Bh
	...	
resp	E9h	C2h

# Exemplo (2) de Passagem por Referência em C++

```
int* metodo(int& a, int* b){  
    a = 5;  
    *b = 6;  
    int* resp = new int[3];  
    resp[0] = 10; resp[1] = 20; resp[2] = 30;  
    return resp;  
}
```

```
int main(int argc, char **argv){  
    int a = 10, b = 25;  
    int* vet = metodo(a, &b);  
    printf("%d %d", a, b);  
    printf("%d %d %d", vet[0], vet[1], vet[2]);  
}
```

E9h	10	20	30
-----	----	----	----

**Tela**

**Memória**

a/a	5	33h
	...	
b	6	51h
	...	
b	51h	7Bh
	...	
resp	E9h	C2h

# Exemplo (2) de Passagem por Referência em C++

```
int* metodo(int& a, int* b){  
    a = 5;  
    *b = 6;  
    int* resp = new int[3];  
    resp[0] = 10; resp[1] = 20; resp[2] = 30;  
    return resp;  
}
```

```
int main(int argc, char **argv){  
    int a = 10, b = 25;  
    int* vet = metodo(a, &b);  
    printf("%d %d", a, b);  
    printf("%d %d %d", vet[0], vet[1], vet[2]);  
}
```

E9h	10	20	30
-----	----	----	----

**Tela**

**Memória**

a/a	5	33h
	...	
b	6	51h
	...	
b	51h	7Bh
	...	
resp	E9h	C2h

# Exemplo (2) de Passagem por Referência em C++

```
int* metodo(int& a, int* b){  
    a = 5;  
    *b = 6;  
    int* resp = new int[3];  
    resp[0] = 10; resp[1] = 20; resp[2] = 30;  
    return resp;  
}
```

```
int main(int argc, char **argv){  
    int a = 10, b = 25;  
    int* vet = metodo(a, &b);  
    printf("%d %d", a, b);  
    printf("%d %d %d", vet[0], vet[1], vet[2]);  
}
```

E9h

10	20	30
----	----	----

Tela

Memória

a 5 33h

b 6 51h

vet E9h 88h

# Exemplo (2) de Passagem por Referência em C++

```
int* metodo(int& a, int* b){  
    a = 5;  
    *b = 6;  
    int* resp = new int[3];  
    resp[0] = 10; resp[1] = 20; resp[2] = 30;  
    return resp;  
}
```

```
int main(int argc, char **argv){  
    int a = 10, b = 25;  
    int* vet = metodo(a, &b);  
    printf("%d %d", a, b);  
    printf("%d %d %d", vet[0], vet[1], vet[2]);  
}
```

E9h    10    20    30

**Tela**

5    6

**Memória**

a    5    33h

b    6    51h

vet    E9h    88h

# Exemplo (2) de Passagem por Referência em C++

```
int* metodo(int& a, int* b){  
    a = 5;  
    *b = 6;  
    int* resp = new int[3];  
    resp[0] = 10; resp[1] = 20; resp[2] = 30;  
    return resp;  
}
```

```
int main(int argc, char **argv){  
    int a = 10, b = 25;  
    int* vet = metodo(a, &b);  
    printf("%d %d", a, b);  
    printf("%d %d %d", vet[0], vet[1], vet[2]);  
}
```

E9h    

10	20	30
----	----	----

**Tela**

5	6
10	20 30

**Memória**

a    

5
---

    33h

...    

...
-----

b    

6
---

    51h

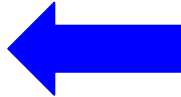
...    

...
-----

vet    

E9h
-----

    88h


- Passagem de Parâmetros por Referência (C++)
- **Estrutura dos Arquivos** 
- Classe Nó
- Classe ArvoreBinaria



# Estrutura de Arquivos

- no.h
- no.cc
- arvorebinaria.h
- arvorebinaria.cc
- principal.cc
- makefile

# Agenda

- Passagem de Parâmetros por Referência (C++)
- Estrutura dos Arquivos
- **Classe Nó** 
- Classe ArvoreBinaria

# Classe Nó em C++

```
//no.h
#include <iostream>

using namespace std;

class No {
public:
    int elemento;
    No *esq, *dir;
    No(int);
};
```

```
//no.cc
#include "no.h"

No::No(int elemento) {
    this->elemento = elemento;
    this->esq = NULL;
    this->dir = NULL;
}
```

# Classe Nó em C++

**OUTRA FORMA DE FAZER  
O CÓDIGO SEM USAR O .H**

```
#include <iostream>

using namespace std;

class No {
public:
    int elemento;
    No *esq, *dir;
    No(int elemento) {
        this->elemento = elemento;
        this->esq = NULL;
        this->dir = NULL;
    }
};
```

# Classe Nó em C++

**TERCEIRA FORMA DE FAZER  
O CÓDIGO SEM USAR O .H**


```
#include <iostream>

using namespace std;

class No {
public:
    int elemento;
    No *esq, *dir;
    No(int);
};

No::No(int elemento) {
    this->elemento = elemento;
    this->esq = NULL;
    this->dir = NULL;
}
```

# Agenda

- Passagem de Parâmetros por Referência (C++)
- Estrutura dos Arquivos
- Classe Nó
- **Classe ArvoreBinaria** 

# Arquivos “arvorebinaria”

```
//arvorebinaria.h
#include "no.h"

class ArvoreBinaria {
private:
    No* raiz;
    bool pesquisar(int, No*);
    void inserir(int, No* &);
    void caminharCentral(No*);
    void caminharPre(No*);
    void caminharPos(No*);
    void remover(int, No* &);
    void maiorEsq(No*, No* &);
public:
    ArvoreBinaria();
    bool pesquisar(int);
    void inserir(int);
    void caminharCentral();
    void caminharPre();
    void caminharPos();
    void remover(int);
};
```

```
//arvorebinaria.cc
#include <err.h>
#include "arvorebinaria.h"

ArvoreBinaria::ArvoreBinaria() {
    raiz = NULL;
}
```

■ ■ ■

# Arquivos “arvorebinaria”

```
//arvorebinaria.h
#include "no.h"

class ArvoreBinaria {
private:
    No* raiz;
    bool pesquisar(int, No*);
    void inserir(int, No* &);
    void caminharCentral(No*);
    void caminharPre(No*);
    void caminharPos(No*);
    void remover(int, No* &);
    void maiorEsq(No*, No* &);
public:
    ArvoreBinaria();
    bool pesquisar(int);
    void inserir(int);
    void caminharCentral();
    void caminharPre();
    void caminharPos();
    void remover(int);
};
```

As implementações apresentadas do inserir em Java/C poderiam ser usadas, contudo, neste material, exploraremos a passagem de parâmetros por referência



# Arquivos “arvorebinaria”

```
//arvorebinaria.h
#include "no.h"

class ArvoreBinaria {
private:
    No* raiz;
    bool pesquisar(int, No*);
    void inserir(int, No* &);
    void caminharCentral(No*);
    void caminharPre(No*);
    void caminharPos(No*);
    void remover(int, No* &);
    void maiorEsq(No*, No* &);
public:
    ArvoreBinaria();
    bool pesquisar(int);
    void inserir(int);
    void caminharCentral();
    void caminharPre();
    void caminharPos();
    void remover(int);
};
```

```
//arvorebinaria.cc

    ■ ■ ■

void ArvoreBinaria::inserir(int x) {
    inserir(x, raiz);
}

void ArvoreBinaria::inserir(int x, No* &i) {
    if (i == NULL) {
        i = new No(x);

    } else if (x < i->elemento) {
        inserir(x, i->esq);

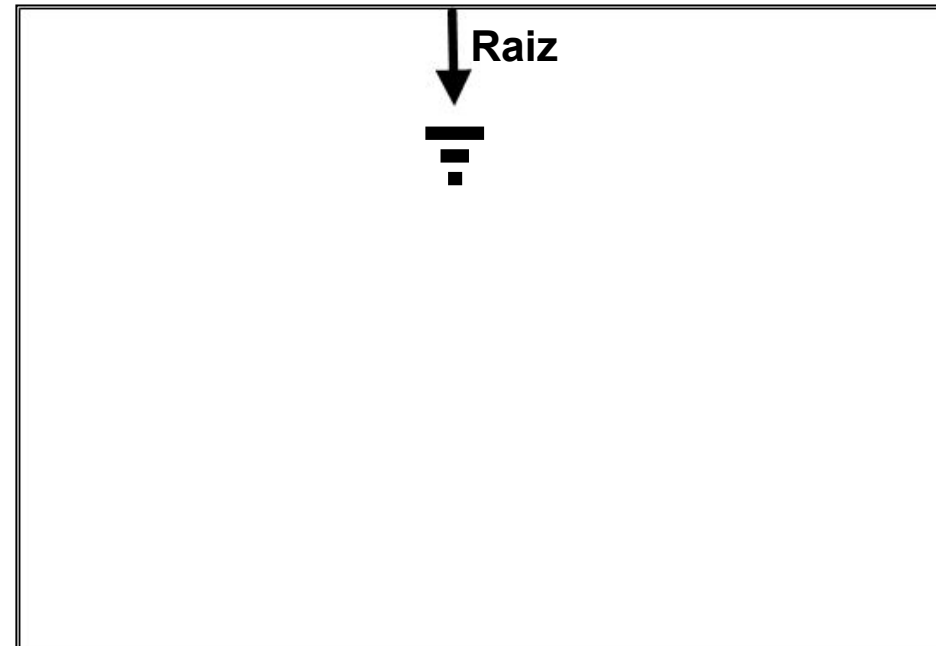
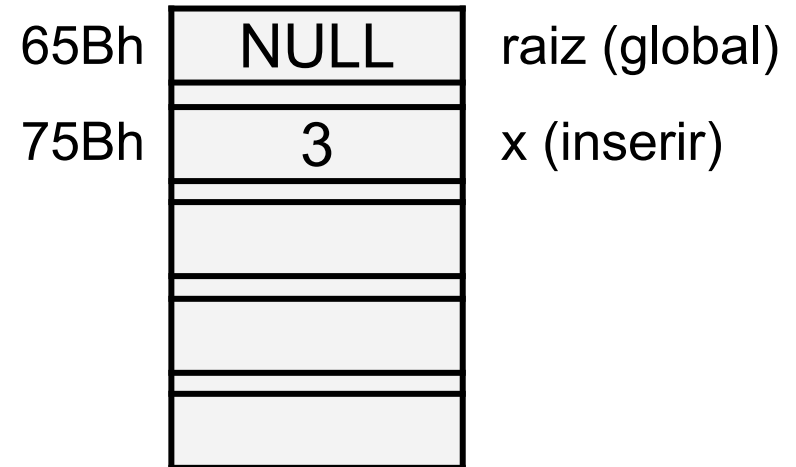
    } else if (x > i->elemento) {
        inserir(x, i->dir);

    } else {
        errx(1, "Erro ao inserir!");
    }
}
```

# Arquivos “arvorebinaria”

//arvorebinaria.cc (supondo inserir o 3)

```
void ArvoreBinaria::inserir(int x) {  
    inserir(x, raiz);  
}  
  
void ArvoreBinaria::inserir(int x, No* &i) {  
    if (i == NULL) {  
        i = new No(x);  
    } else if (x < i->elemento) {  
        inserir(x, i->esq);  
    } else if (x > i->elemento) {  
        inserir(x, i->dir);  
    } else {  
        errx(1, "Erro ao inserir!");  
    }  
}
```

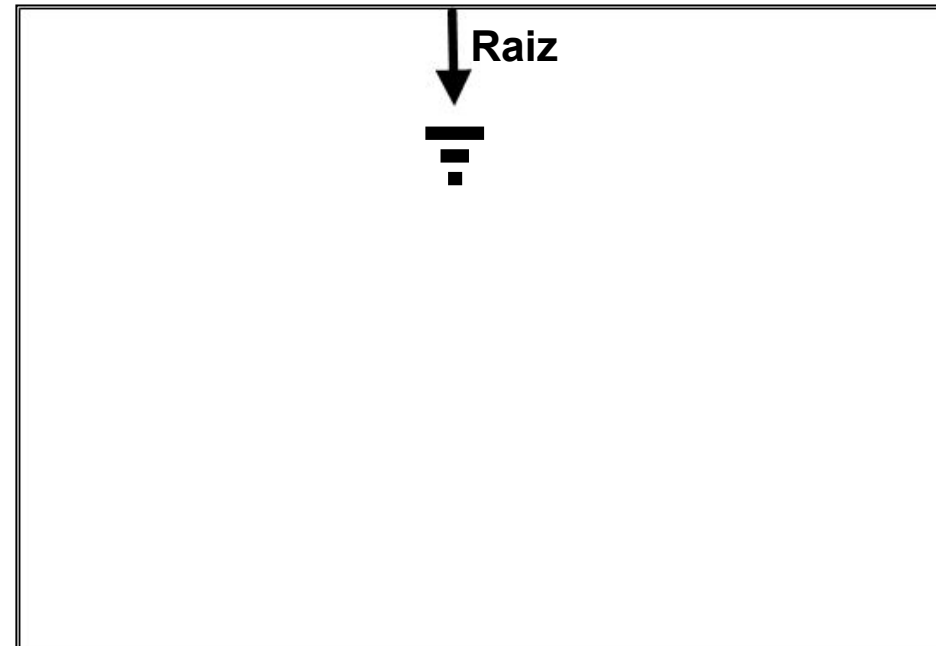
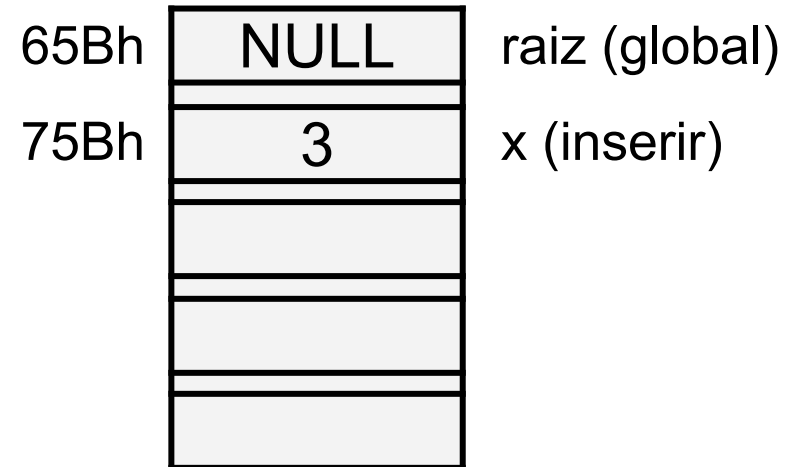


# Arquivos “arvorebinaria”

```
//arvorebinaria.cc (supondo inserir o 3)
```

```
void ArvoreBinaria::inserir(int x) {  
    inserir(x, raiz);  
}
```

```
void ArvoreBinaria::inserir(int x, No* &i) {  
    if (i == NULL) {  
        i = new No(x);  
    } else if (x < i->elemento) {  
        inserir(x, i->esq);  
    } else if (x > i->elemento) {  
        inserir(x, i->dir);  
    } else {  
        errx(1, "Erro ao inserir!");  
    }  
}
```



# Arquivos “arvorebinaria”

//arvorebinaria.cc (supondo inserir o 3)

```
void ArvoreBinaria::inserir(int x) {  
    inserir(x, raiz);  
}
```

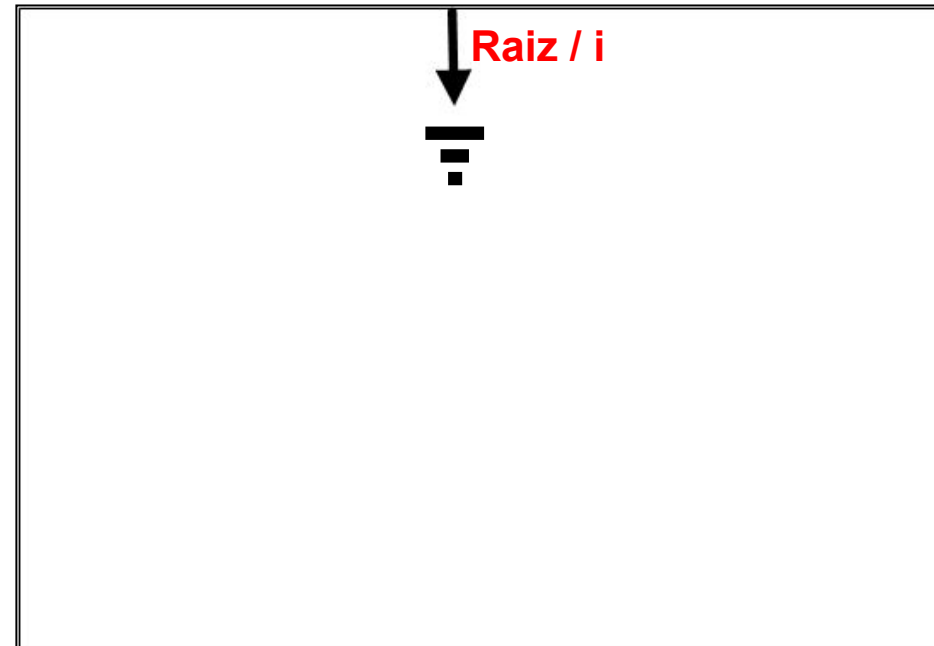
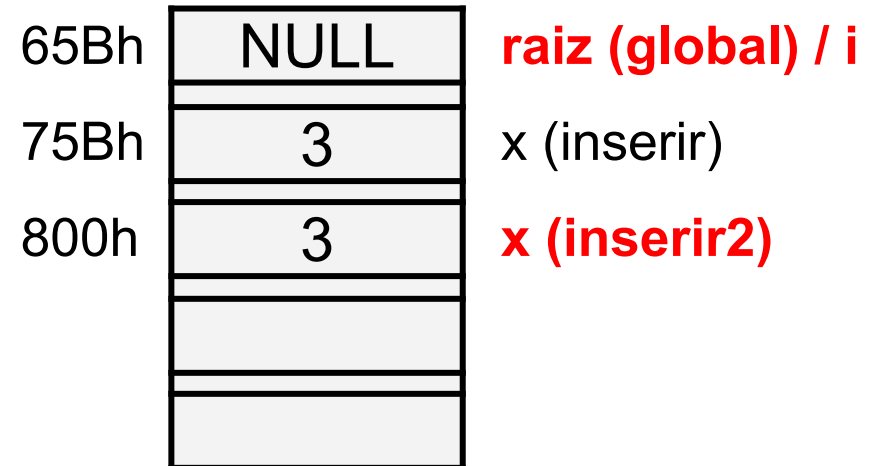
```
void ArvoreBinaria::inserir(int x, No* &i) {
```

```
    if (i == NULL) {  
        i = new No(x);
```

```
    } else if (x < i->elemento) {  
        inserir(x, i->esq);
```

```
    } else if (x > i->elemento) {  
        inserir(x, i->dir);
```

```
    } else {  
        errx(1, "Erro ao inserir!");  
    }  
}
```



# Arquivos “arvorebinaria”

```
//arvorebinaria.cc (supondo inserir o 3)
```

```
void ArvoreBinaria::inserir(int x) {  
    inserir(x, raiz);  
}
```

```
void ArvoreBinaria::inserir(int x, No* &i) {
```

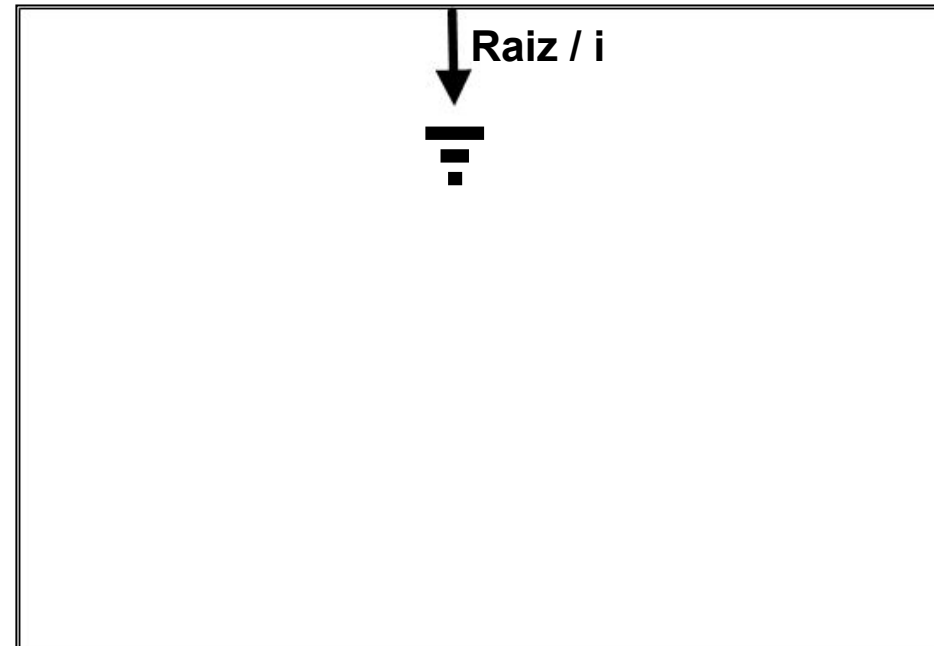
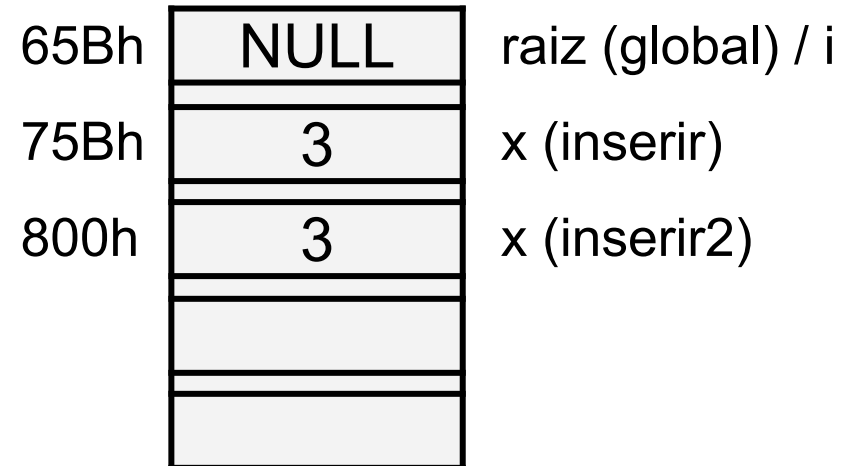
```
    if (i == NULL) {
```

```
        i = new No(x);
```

```
    } else if (x < i->elemento) {  
        inserir(x, i->esq);
```

```
    } else if (x > i->elemento) {  
        inserir(x, i->dir);
```

```
    } else {  
        errx(1, "Erro ao inserir!");  
    }  
}
```



# Arquivos “arvorebinaria”

```
//arvorebinaria.cc (supondo inserir o 3)
```

```
void ArvoreBinaria::inserir(int x) {  
    inserir(x, raiz);  
}
```

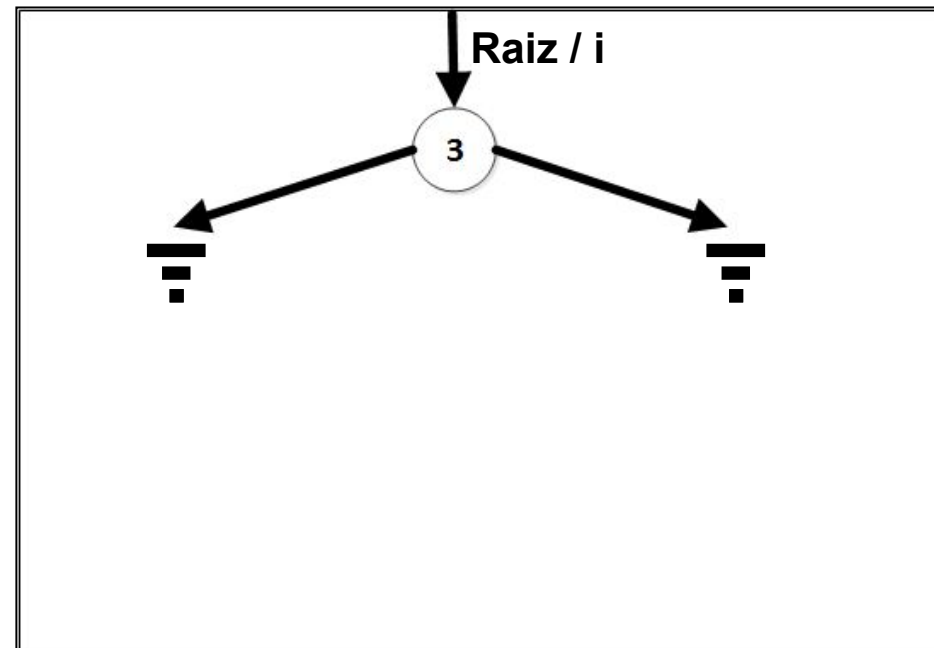
```
void ArvoreBinaria::inserir(int x, No* &i) {  
    if (i == NULL) {  
        i = new No(x);
```

```
    } else if (x < i->elemento) {  
        inserir(x, i->esq);
```

```
    } else if (x > i->elemento) {  
        inserir(x, i->dir);
```

```
    } else {  
        errx(1, "Erro ao inserir!");  
    }  
}
```

65Bh	922h	raiz (global) / i
75Bh	3	x (inserir)
800h	3	x (inserir2)
922h	3/null/null	(novoNo)



# Arquivos “arvorebinaria”

//arvorebinaria.cc (supondo inserir o 3)

```
void ArvoreBinaria::inserir(int x) {  
    inserir(x, raiz);  
}
```

```
void ArvoreBinaria::inserir(int x, No* &i) {  
    if (i == NULL) {  
        i = new No(x);
```

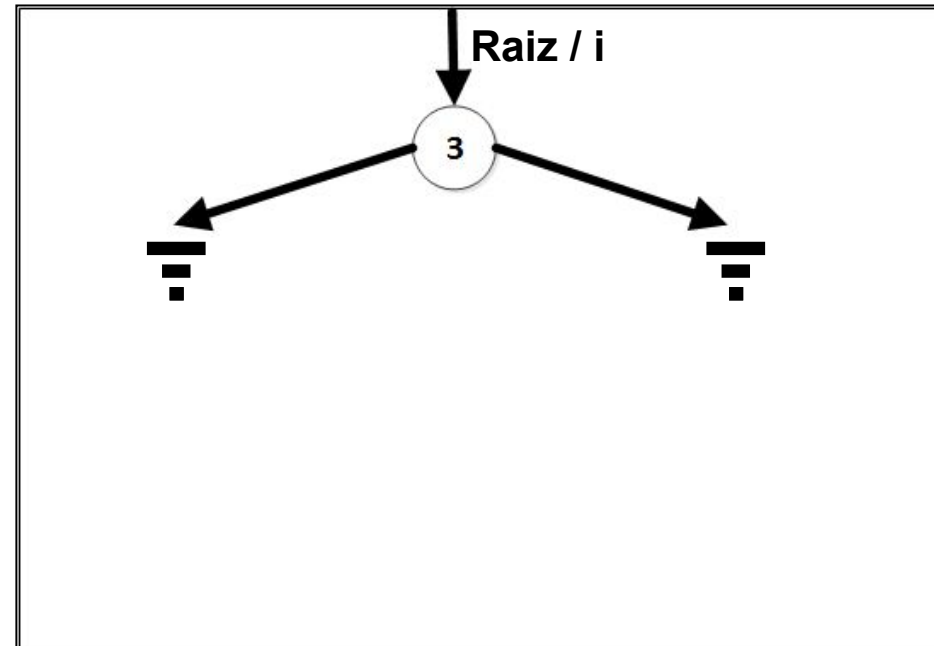
```
    } else if (x < i->elemento) {  
        inserir(x, i->esq);
```

```
    } else if (x > i->elemento) {  
        inserir(x, i->dir);
```

```
    } else {  
        errx(1, "Erro ao inserir!");  
    }
```

```
}
```

65Bh	922h	raiz (global) / i
75Bh	3	x (inserir)
800h	3	x (inserir2)
922h	3/null/null	(novoNo)



# Arquivos “arvorebinaria”

```
//arvorebinaria.cc (supondo inserir o 3)
```

```
void ArvoreBinaria::inserir(int x) {  
    inserir(x, raiz);  
}
```

```
void ArvoreBinaria::inserir(int x, No* &i) {  
    if (i == NULL) {  
        i = new No(x);  
    } else if (x < i->elemento) {  
        inserir(x, i->esq);  
    } else if (x > i->elemento) {  
        inserir(x, i->dir);  
    } else {  
        errx(1, "Erro ao inserir!");  
    }  
}
```

65Bh	922h	raiz (global) / i
75Bh	3	x (inserir)
800h	3	x (inserir2)
922h	3/null/null	(novoNo)

