

40. Desenvolver um programa na sua linguagem de programação favorita, que apresente a ordem de avaliação dos operandos utilizada pela linguagem de programação selecionada. Explicar o que está acontecendo no código desenvolvido, e se há ou não a ocorrência de "efeito colateral" na linguagem.

Linguagem de Programação Pascal

```
type PInt = ^integer;
```

```
var x, y : integer;
```

```
function sub(a, b : PInt) : integer;
```

```
  begin
```

```
    a^ := a^ + 1;
```

```
    b^ := b^ + 2;
```

```
    sub := a^ + b^;
```

```
  end;
```

```
begin
```

```
  x := 2;
```

```
  y := 4;
```

```
  writeln(x - y + sub(@x, @y));
```

```
  // 2 - 4 + (3 + 6) = 7
```

```
  // sem ocorrência de efeito colateral
```

```
  x := 2;
```

```
  y := 4;
```

```
  writeln(x + sub(@x, @y) - y);
```

```
  // 2 + (3 + 6) - 6 = 5
```

```
  // ocorrência de efeito colateral no terceiro operando
```

```
  x := 2;
```

```
  y := 4;
```

```
  writeln(sub(@x, @y) + x - y);
```

```
  // (3 + 6) + 3 - 6 = 6
```

```
  // ocorrência de efeito colateral no segundo e terceiro operando
```

```
  // Ordem de avaliação dos operandos é da esquerda para a direita
```

```
end.
```

Linguagem de Programação C

```
#include <stdio.h>
```

```
int sub(int *a, int *b)
{
    *a = *a + 1;
    *b = *b + 2;
    return *a + *b;
}
```

```
int main()
{
    int x = 2, y = 4;
    printf("%d\n", x - y + sub(&x, &y));
    // 2 - 4 + (3 + 6) = 7
    // sem ocorrência de efeito colateral

    x = 2, y = 4;
    printf("%d\n", x + sub(&x, &y) - y);
    // 3 + (3 + 6) - 6 = 6
    // ocorrência de efeito colateral no segundo e terceiro operando

    x = 2, y = 4;
    printf("%d\n", sub(&x, &y) + x - y);
    // (3 + 6) + 3 - 6 = 6
    // ocorrência de efeito colateral no segundo e terceiro operando

    // Ordem de avaliação dos operandos é da esquerda para a direita,
    // mas a função tem precedência sobre os operandos escalares

    return 0;
}
```

Linguagem de Programação Python

```
x = 2  
y = 4
```

```
def sub():  
    global x  
    x = x + 1  
  
    global y  
    y = y + 2  
  
    return x + y
```

```
print (x - y + sub())  
# 2 - 4 + (3 + 6) = 7  
# sem ocorrência de efeito colateral
```

```
x = 2  
y = 4  
print (x + sub() - y)  
# 2 + (3 + 6) - 6 = 5  
# ocorrência de efeito colateral no terceiro operando
```

```
x = 2  
y = 4  
print (sub() + x - y)  
# (3 + 6) + 3 - 6 = 6  
# ocorrência de efeito colateral no segundo e terceiro operando  
  
# Ordem de avaliação dos operandos é da esquerda para a direita
```