

- 3) Acompanhe a execução passo a passo do programa abaixo utilizando o depurador do Visual Studio para tentar descobrir por que a variável `b` tem valor igual a zero no fim da execução.

```
#include <iostream>
using namespace std;

int main()
{
    int a, b, c = 0;

    a = 1;
    b = 2;
    a = c;
    b = a;

    cout << b << endl;
    return 0;
}
```

Quais os valores das variáveis `a` e `b` após executar a linha abaixo? Todas as variáveis iniciam com valor zero?

```
int a, b, c = 0;
```

Apenas a variável “`c`” inicia com valor 0, “`a`” e “`b`” iniciam sem valor, ou com valor “-858993460”.

Se o código acima for alterado de forma que a variável `c` não receba o valor zero, como mostrado abaixo, o programa ainda compila corretamente? Se não, veja a mensagem de erro do compilador e explique o porquê do erro.

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

Não, o compilador apresenta erro na seguinte linha:

```
a = c;
```

O erro acontece pois a variável “`a`” está recebendo o valor de “`c`”, que não tem valor.

- 4) Ao iniciar a depuração do código abaixo, você entende por que o depurador salta imediatamente para a linha “a = 1;”? Isso acontecia com o programa da questão anterior? Qual a explicação para essa diferença?

```
#include <iostream>
using namespace std;

int main()
{
    int a, b, c;

    a = 1;
    b = 2;
    c = 3;

    return 0;
}
```

Acredito que o depurador salta para essa linha pois, na linha anterior, nenhuma operação é realizada. Isso não ocorria no programa da questão anterior, pois a primeira linha da função main realizava a operação de atribuir o valor 0 à variável “c”.