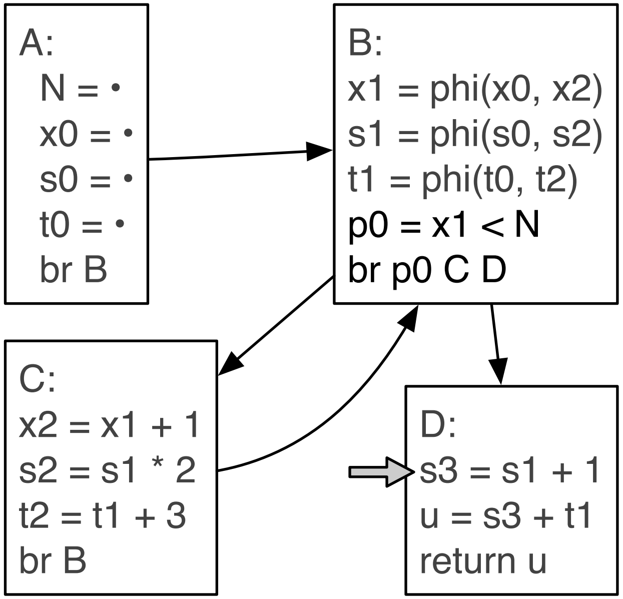
Consider the program below. The arrow indicates a particular instruction of that program (a *slice criterion*). Would you be able to produce the minimum subprogram that is necessary to execute that instruction?

```
01  int foo(int N) {
02    int x = 0;
03    int s = 1;
04    int t = 0;
05    while (x < N) {
06      x += 1;
07      s *= 2;
08      t += 3;
09    }
10    s = s + 1 ⇦
11    u = s = t
12    return u;
13  }
```



```
A:
  N = •
  x0 = •
  s0 = •
  t0 = •
  br B

B:
x1 = phi(x0, x2)
s1 = phi(s0, s2)
t1 = phi(t0, t2)
p0 = x1 < N
br p0 C D

C:
x2 = x1 + 1
s2 = s1 * 2
t2 = t1 + 3
br B

D:
s3 = s1 + 1
u = s3 + t1
return u
```

**Figure 1**: A program in source code and in SSA-form. Let the assignment at Line 10 be a "slice criterion".

1. Can you think about a way to separate the minimum number of program statements that yields the right execution of the assignment s = s + 1?

```
01  int foo(int N) {
02    int x = 0;
03    int s = 1;
04    int t = 0;
05    while (x < N) {
06      x += 1;
07      s *= 2;
08      t += 3;
09    }
10    s = s + 1
11    u = s = t
12    return u;
13  }
```

UFMG

DCC

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Compilers Laboratory

**Figure 2**: The minimum subprogram that always executes the assignment s = s + 1 as the original program.

2. Notice that whichever technique you might come up with to answer Q1 must take into consideration the fact that there is no data dependencies between the instruction in Line 10 of Figure 1 and the loop. So, how can you relate the instruction at Line 10 and the loop? The number of iterations in the loop is essential to compute the right value of ’s’.

3. This minimum program that ensures the right execution of a statement is the program slice with regards to that statement. Is the analysis that computes the program slice may or must? Forward or backward?