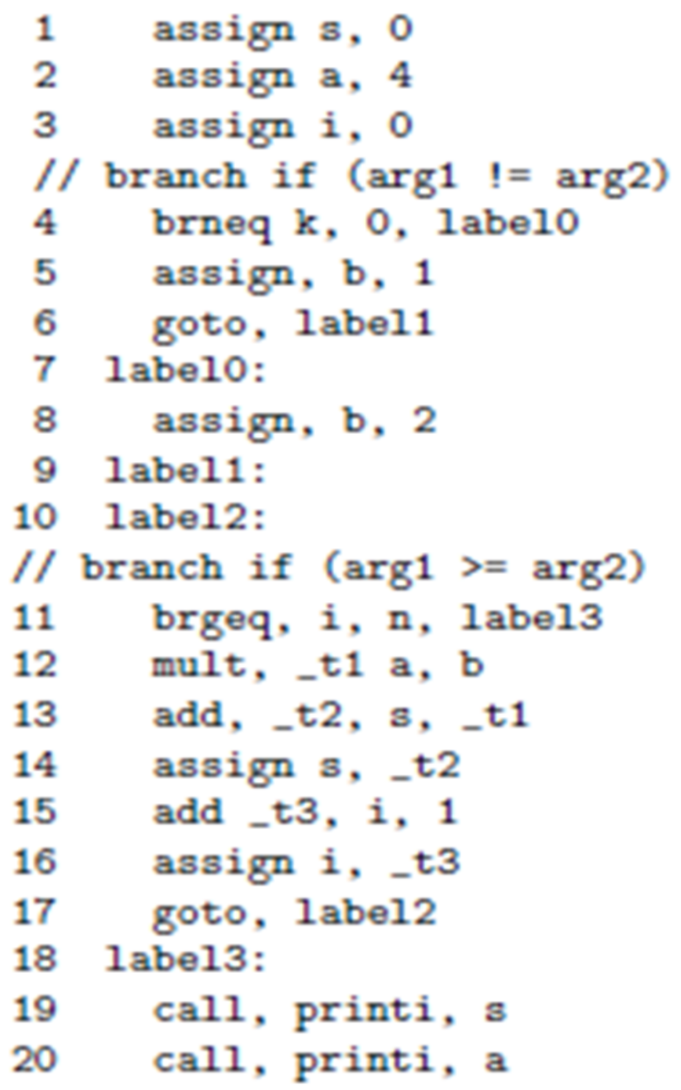
Worksheet-2 Solution

(From Lecture 2 given on 01/09/2019)

Q1

- Construct a Control Flow Graph for the IR (Intermediate Representation) segment shown below, and draw it on the right of the IR. Each vertex can be a single IR instruction, or a basic block containing of a straight-line sequence of multiple IR instructions.



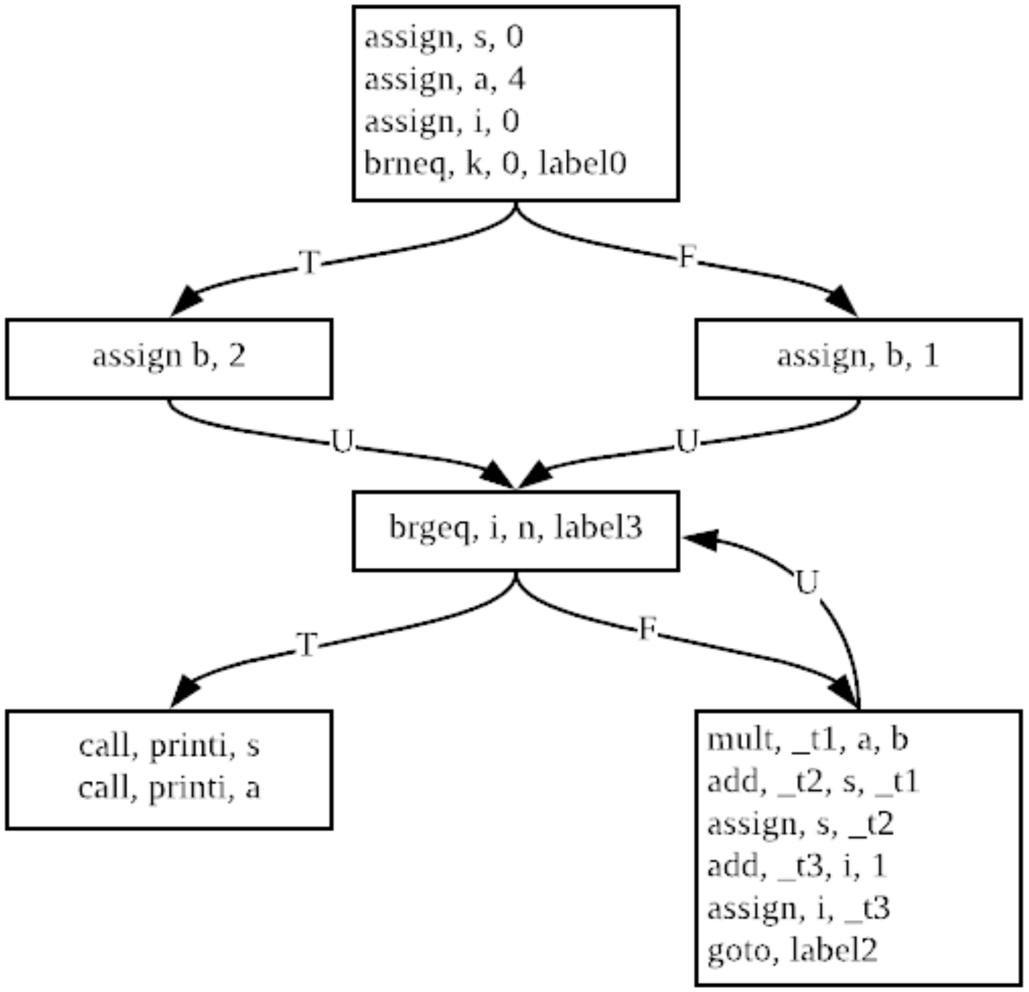
```
1    assign s, 0
2    assign a, 4
3    assign i, 0
// branch if (arg1 != arg2)
4    brneq k, 0, label0
5    assign, b, 1
6    goto, label1
7  label0:
8    assign, b, 2
9  label1:
10 label2:
// branch if (arg1 >= arg2)
11   brgeq, i, n, label3
12   mult, _t1 a, b
13   add, _t2, s, _t1
14   assign s, _t2
15   add _t3, i, 1
16   assign i, _t3
17   goto, label2
18 label3:
19   call, printi, s
20   call, printi, a
```

Two answers possible for Q1

- It is a <u>design choice</u> whether **label**s should be considered as “no-op IR instructions” in CFG vertices, or should be excluded from CFG vertices
  - Just like basic block granularity (minimal vs. maximal) is a design choice when implementing CFGs
- Resulting Control Flow Graphs can be slightly different, depending on the assumption made for labels.
- Most students answered correctly in either case.

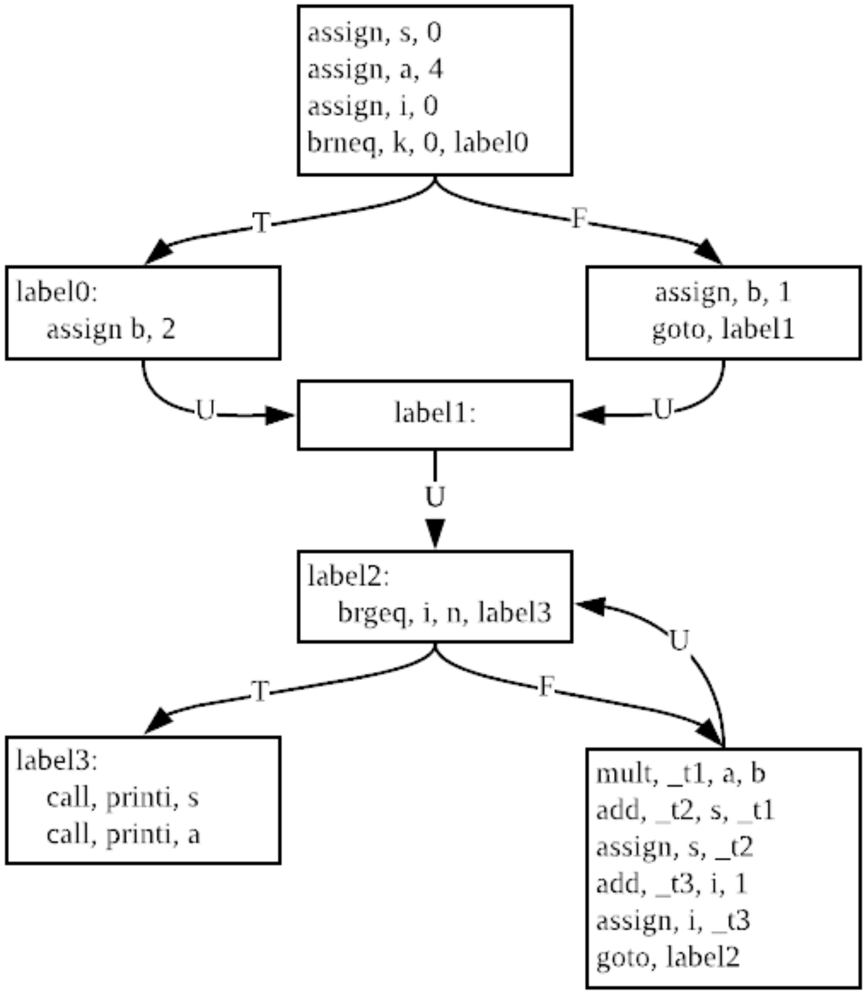
Q1 Sample Solution1: labels are not instructions

```
  assign s, 0
  assign a, 4
  assign i, 0
// branch if (arg1 != arg2)
  brneq k, 0, label0
  assign, b, 1
  goto, label1
label0:
  assign, b, 2
label1:
label2:
// branch if (arg1 >= arg2)
  brgeq, i, n, label3
  mult, _t1 a, b
  add, _t2, s, _t1
  assign s, _t2
  add _t3, i, 1
  assign i, _t3
  goto, label2
label3:
  call, printi, s
  call, printi, a
```



Q1 Sample Solution2: labels are no-op instructions

```
   assign s, 0
   assign a, 4
   assign i, 0
// branch if (arg1 != arg2)
   brneq k, 0, label0
   assign, b, 1
   goto, label1
label0:
   assign, b, 2
label1:
label2:
// branch if (arg1 >= arg2)
   brgeq, i, n, label3
   mult, _t1 a, b
   add, _t2, s, _t1
   assign s, _t2
   add _t3, i, 1
   assign i, _t3
   goto, label2
label3:
   call, printi, s
   call, printi, a
```



```
assign, s, 0
assign, a, 4
assign, i, 0
brneq, k, 0, label0
```

```
label0:
   assign b, 2
```

```
assign, b, 1
goto, label1
```

```
label1:
```

```
label2:
   brgeq, i, n, label3
```

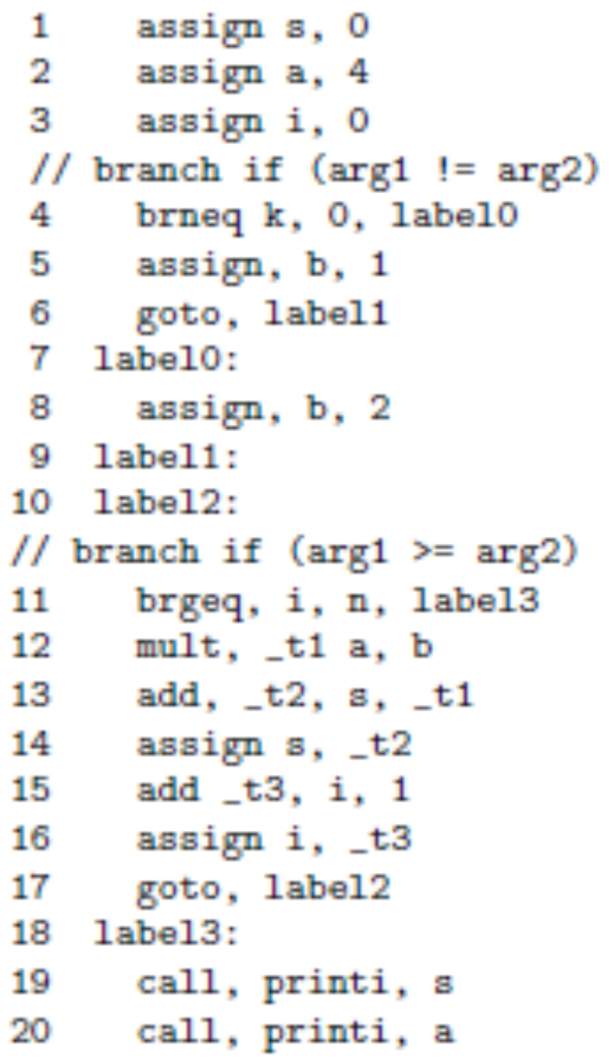
```
label3:
   call, printi, s
   call, printi, a
```

```
mult, _t1, a, b
add, _t2, s, _t1
assign, s, _t2
add, _t3, i, 1
assign, i, _t3
goto, label2
```

Q2, Q3

- Q2
  Which uses of ***a*** are reached by the def of ***a*** in line 2?

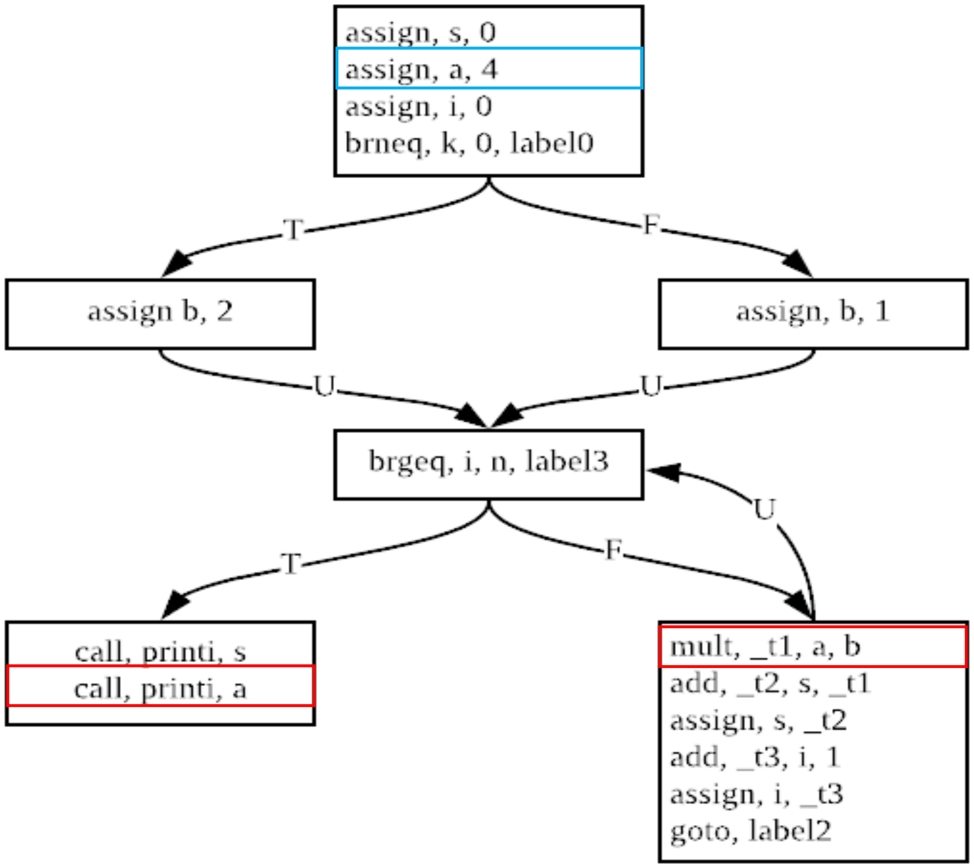
- Q3
  Which defs of ***b*** reach the use of ***b*** in line 12?



```
 1    assign s, 0
 2    assign a, 4
 3    assign i, 0
// branch if (arg1 != arg2)
 4    brneq k, 0, label0
 5    assign, b, 1
 6    goto, label1
 7  label0:
 8    assign, b, 2
 9  label1:
10  label2:
// branch if (arg1 >= arg2)
11    brgeq, i, n, label3
12    mult, _t1 a, b
13    add, _t2, s, _t1
14    assign s, _t2
15    add _t3, i, 1
16    assign i, _t3
17    goto, label2
18  label3:
19    call, printi, s
20    call, printi, a
```

Q2 Solution

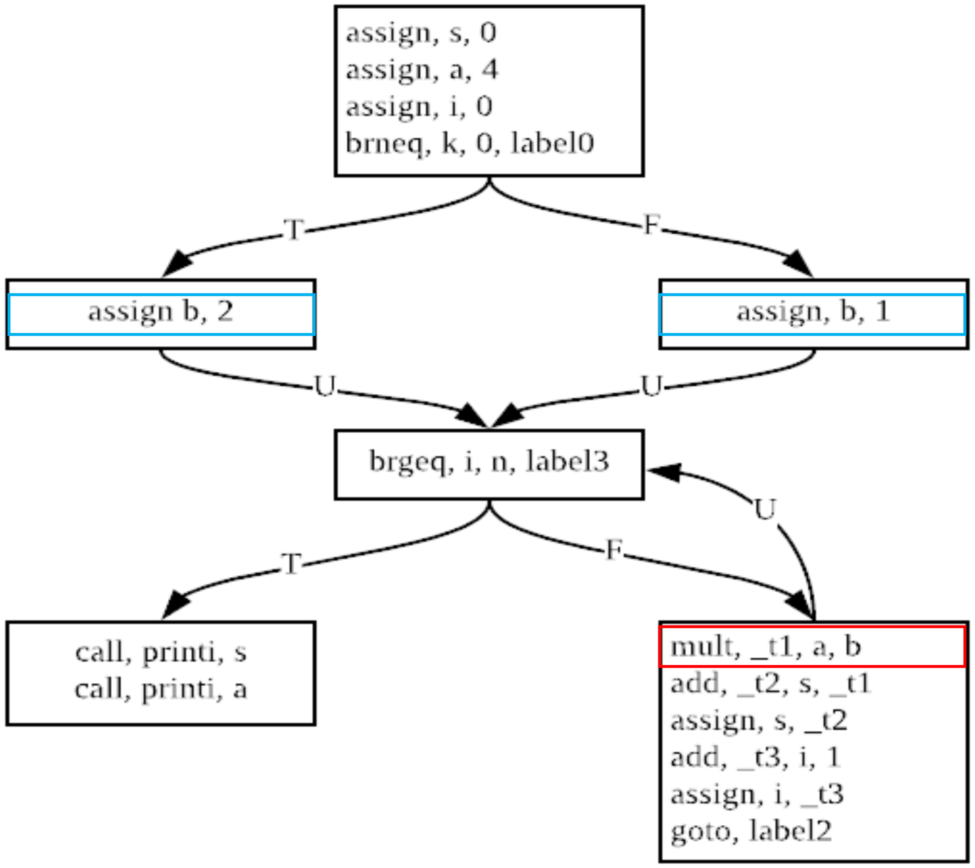
```
1     assign s, 0
2     assign a, 4
3     assign i, 0
// branch if (arg1 != arg2)
4     brneq k, 0, label0
5     assign, b, 1
6     goto, label1
7   label0:
8     assign, b, 2
9   label1:
10  label2:
// branch if (arg1 >= arg2)
11    brgeq, i, n, label3
12    mult, _t1 a, b
13    add, _t2, s, _t1
14    assign s, _t2
15    add _t3, i, 1
16    assign i, _t3
17    goto, label2
18  label3:
19    call, printi, s
20    call, printi, a
```



In line 2, variable ‘a’ is defined as 4. <u>The definition of ‘a’ in line 2 reaches line **12 & 20** without any intervening ‘def of a’.</u>

Q3 Solution

```
1     assign s, 0
2     assign a, 4
3     assign i, 0
// branch if (arg1 != arg2)
4     brneq k, 0, label0
5     assign, b, 1
6     goto, label1
7   label0:
8     assign, b, 2
9   label1:
10  label2:
// branch if (arg1 >= arg2)
11    brgeq, i, n, label3
12    mult, _t1 a, b
13    add, _t2, s, _t1
14    assign s, _t2
15    add _t3, i, 1
16    assign i, _t3
17    goto, label2
18  label3:
19    call, printi, s
20    call, printi, a
```



```
assign, s, 0
assign, a, 4
assign, i, 0
brneq, k, 0, label0
```

```
assign b, 2
```

```
assign, b, 1
```

```
brgeq, i, n, label3
```

```
call, printi, s
call, printi, a
```

```
mult, _t1, a, b
add, _t2, s, _t1
assign, s, _t2
add, _t3, i, 1
assign, i, _t3
goto, label2
```

There is no intervening ‘def of b’ in the control flow etween line 5 and line 12. Same for line 8 and line 12. The defs of b in lines **5 & 8** both reach the use of b in ine 12.

Comments about Q2, Q3

- Most students answered these questions correctly.
- Some students were not sure about the meaning of the terms **‘def’** and **‘use’**.

  ‘def’ :
  a write operation on a variable (short for “definition”)

  ‘use’ :
  a read operation on a variable