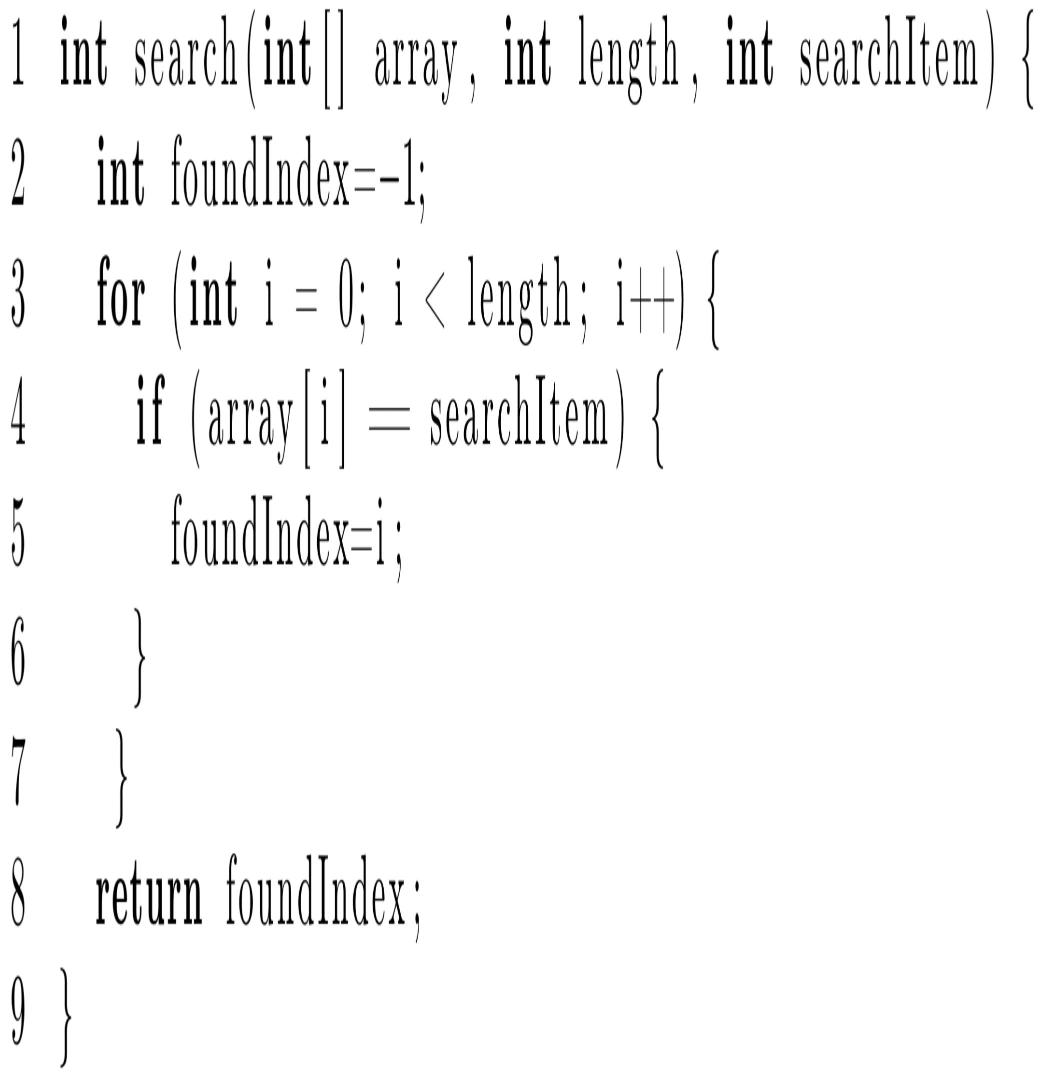
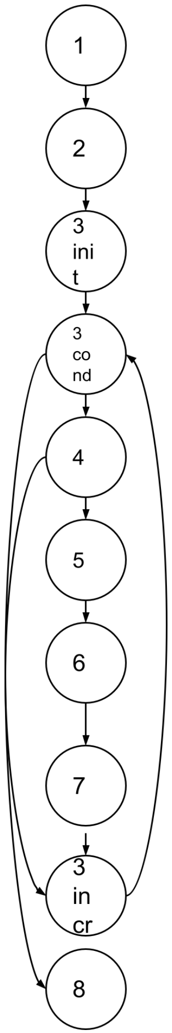
Assignment 5

Print as PDF: use A4, landscape, scale: custom number 80%, narrow margins! Submit PDF to Gradescope.



```
1 int search(int[] array, int length, int searchItem) {
2   int foundIndex=-1;
3   for (int i = 0; i < length; i++) {
4     if (array[i] == searchItem) {
5       foundIndex=i;
6     }
7   }
8   return foundIndex;
9 }
```

1. CFG



7. def/use annotations

	array[]	length	searchItem	foundIndex	i
1 => Note 2	def	def	def		
2 => Note 3 init				def	
Note 3 init => Note 3 cond					def
Note 3 cond => Note 4		p			p
Note 4 => Note 7	p		p		p
Note 4 => Note 5	p		p		p
Note 5			def		c
Note 6					
Note 7					
Note 3 incr					c def
Note 8			c		

Double click to edit

I'm assuming that line 4 is like this: array[i] == searchItem

To copy the above CFG drawing, open it in the drawing editor, copy it there (CTRL-A, CTRL-C), (save&)close the editor,
create a new drawing (Insert->Drawing) and paste (CTRL-C) in the new drawing editor the old,copied drawing.

2. Paths statement coverage

Path 1	Path 2	Path 3
1	1	1
2	2	2
3 init	3 init	3 init
3 cond	3 cond	3 cond
4	4	4
5	5	3 incr
6	6	3 cond
7	7	4
3 incr	3 incr	5
3 cond	3 cond	6
8	4	7
	3 incr	3 incr
	3 cond	3 cond
	8	8

3. Additional paths branch coverage

Path i	Path ii	Path iii
1	1	1
2	2	2
3 init	3 init	3 init
3 cond	3 cond	3 cond
8	4	4
	3 incr	3 incr
	3 cond	3 cond
	8	4
		3 incr
		3 cond
		8

4. Upper bound #paths branch coverage

The upper bound is: 11 - 10 + 2 = **3**

7. Further paths needed for all-p uses coverages

Further paths needed: Yes

Justification:

Let's say for example that the array is of length 10.000 with values [0..9.999] and the attribute int length is 10.000. Let's also assume that SearchItem is the integer 10.000. Since this is a linear Search we would have to go through the loop 10.000 times.

5. Additional paths for boundary-interior path coverage

Exterior path	Boundary path	Interior path
1	1	1
2	2	2
3 init	3 init	3 init
3 cond	3 cond	3 cond
8	4	4
	5	5
	6	6
	7	7
	3 incr	3 inr
	3 cond	3 cond
	8	4
		7
		3 incr
		3 cond
		8

6. Number of paths as function of length

numberOfPaths(length):= length-3*4 (Replace)