

University of California, Berkeley – College of Engineering

Department of Electrical Engineering and Computer Sciences

Fall 2011

Instructor: Dan Garcia

2011-10-25

CS10 PAPER MIDTERM

<i>Last Name</i>	
<i>First Name</i>	
<i>Student ID Number</i>	
<i>cs10- Login First Letter</i>	a b c d e f g h i j k l m
<i>cs10- Login Last Letter</i>	a b c d e f g h i j k l m n o p q r s t u v w x y z
<i>The name of your LAB TA (please circle)</i>	Aijia Glenn Luke Navin Rabbit Samir
<i>Name of the person to your Left</i>	
<i>Name of the person to your Right</i>	
<i>All my work is my own. I had no prior knowledge of the exam contents nor will I share the contents with others in CS10 who have not taken it yet. (please sign)</i>	

Instructions

- Don't Panic!
- This booklet contains 6 pages including this cover page. Put all answers on these pages; don't hand in any stray pieces of paper.
- Please turn off all pagers, cell phones and beepers. Remove all hats and headphones.
- You have 110 minutes to complete this exam. The midterm is closed book, no computers, no PDAs, no cell phones, no calculators, but you are allowed two double-sided sets of notes. There may be partial credit for incomplete answers; write as much of the solution as you can. When we provide a blank, please fit your answer within the space provided.

Question	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Online	Total
Points	2	2	2	2	2	2	2	2	6	2	4	10	8	4	10	60
Score																

If you can draw and you have time, feel free to doodle all over this front page!

Short-answer Questions (2 pts each, the last one 6 pts)

Question 1: We almost always want our algorithms to be correct. When would we soften that requirement? (i.e., what other concern might we have so that we would happily accept answers that weren't 100% correct?)

Question 2: Mobile computing users claim that it isn't *Moore's law* that affects them the most, but *Koomey's law*, which found something very important doubled every 18 months (unabated, since 1950). What was it?

Question 3: What was the unintended consequence of the shift to multiple-choice (computer-graded) tests?

Question 4: Sir Ken Robinson talked about "changing educational paradigms" ... name one paradigm.

Question 5: What is so great about *Creative Commons*? (i.e., what does it allow an author to do?)

Question 6: The *Connected* movie ends with a powerful quote: "*For centuries we've been declaring independence. Perhaps it's time we declared our interdependence*". What dominant technology has brought us to this point, and once we've made this new declaration, what is the movie suggesting we do next?

Question 7: In 1997, the MA Group Insurance Commission (GIC) released a 135,000-patient dataset, but made sure to "de-identify" it, removing names, addresses, SSNs, and telephone numbers. What happened next?

Question 8: For *four thousand years*, cryptography was done a certain way, until the 1970s. What changed?

Question 9: Match the person with what made them famous. Not all numbers need to be used.

- | | |
|------------------------|--|
| a) Vint Cerf | 1) Invented the first laserprinter |
| b) John Warnock | 2) Built first web server in 1990, inventor of the world wide web |
| c) Doug Englebart | 3) Invented Postscript, a language used for compactly specifying documents |
| d) Sir Tim Berners-Lee | 4) Showed how to hack a Diebold voting machine to produce improper results |
| e) Judah Schwartz | 5) Wrote VisiCalc, one of the first PC spreadsheets |
| f) Harri Hursti | 6) One of the inventors of the transistor. |
| | 7) Gave "mother of all demos", showing first mouse and videoconference |
| | 8) Created one of the first popular web browsers |
| | 9) One of the developers of TCP, an important piece of Internet Protocol Suite |
| | 10) Came up with Tools-vs-Microworlds-vs-Courseware theory |

Login: cs10-_____

Question 10: A classy question! (2 pts)

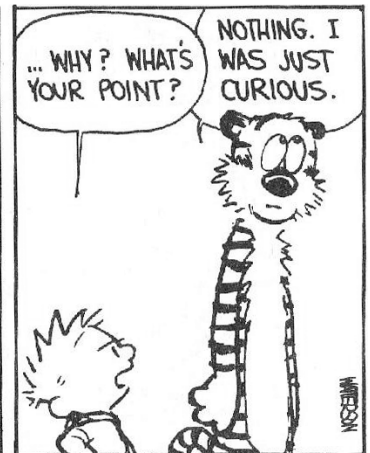
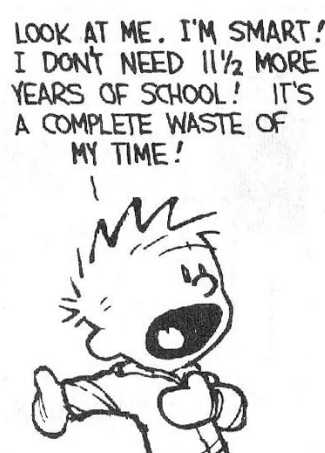
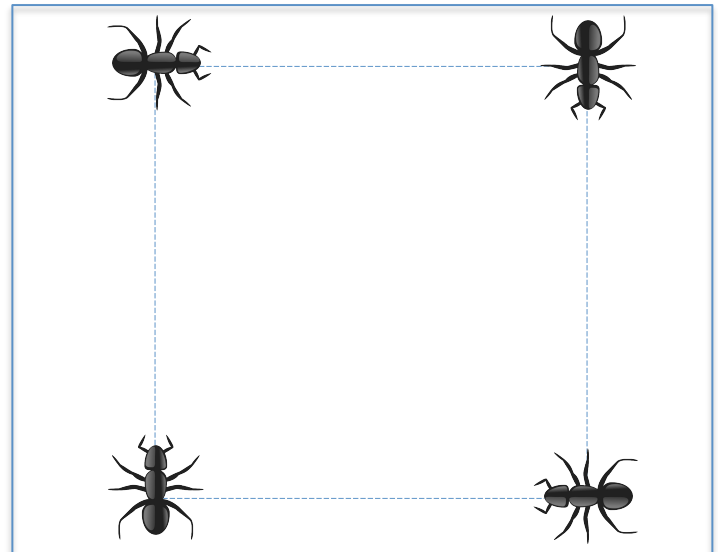
Draw the **CLASSES** list so that the following code returns **CS10**.

item 1 of item 2 of CLASSES

Question 11: I've got ants in my BYOB! (4 pts)

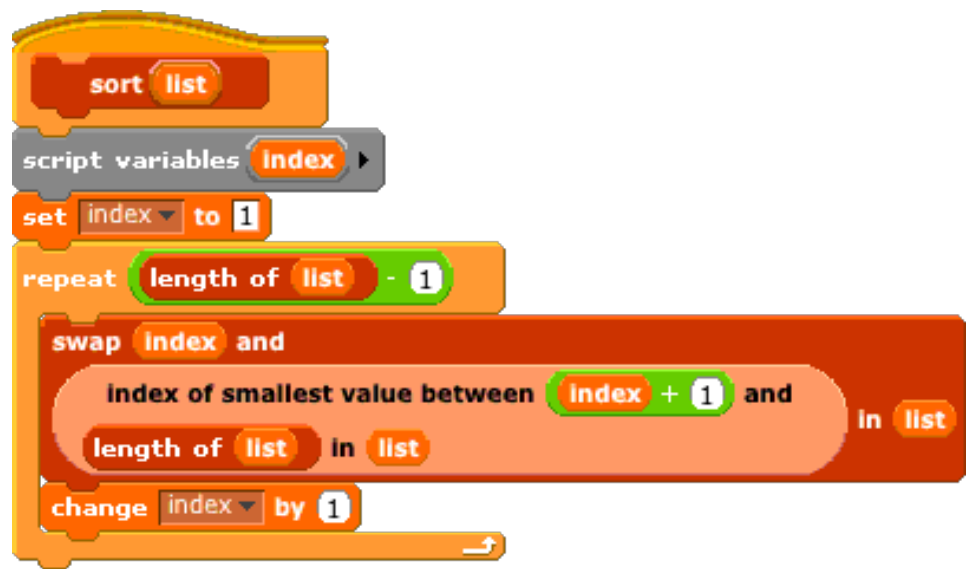
You place 4 ants (each a different sprite, with the “pen” in the center of the ant) on the four corners of an imaginary square shown in the picture below, each facing the ant to their left. Each runs the same command when the green flag is clicked (the block **my-ant** reports the ant you're chasing).

Draw what lines you would see on the stage after the program stops.



Question 12: *Some sort of problem with my code...* (10 pts)

We would like to take an unsorted list and *sort* it (into *increasing* order – the smallest element at index 1, the second smallest at index 2, etc). We’ve tried to write code to do this for us, but we believe it has a bug. The idea (that would work if we could get the details right) is to divide the list into two parts: the sub-list of items already sorted, which is built up from left to right and is found at the beginning, and the sub-list of items remaining to be sorted, occupying the remainder of the array. This is called “selection sort”.



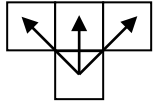
We use two bug-free helper blocks described below:

Helper block	Description
	Search the list for the smallest value between the left-index and the right-index (<i>inclusive</i> , meaning <i>including</i> the elements living at the left and right indices) and report the index of the smallest value in that range
	Swap the elements at left-index and right-index in the list . The length of the list remains unchanged.

- a) What is the running time of `sort`? _____
- b) If `list` is (4 3 2 5 1), what is `list` after `sort(list)`? _____
- c) Most of the time `sort(list)` *doesn't* leave `list` sorted. Describe what an 100-element `list` (of all the numbers 1 through 100 in some order) would look like so that `sort(list)` does leave `list` sorted.
- d) Briefly describe the single, very small change needed to fix the bug.

Question 13: Strawberry Fields Forever... (8 pts)

Strawberry plants are funny. Every year they send “runners” to their left and right neighbors, which take seed and become an entirely new strawberry plant the next year. We’d like to model this process and count how many strawberry plants we’ll have in our garden (that we’ve divided into columns, like the number line) starting from a single strawberry plant in column 0 in year 1, the top row. All other numbers in the top row are 0 (no other strawberry plants). The number in every subsequent row is the sum of the three numbers directly above it, to the above left and to the above right as shown below. We’ve filled in the table for years 1 through 5:



		...	-4	-3	-2	-1	0	1	2	3	4	...
Y E A R	1	...	0	0	0	0	1	0	0	0	0	...
	2	...	0	0	0	1	1	1	0	0	0	...
	3	...	0	0	1	2	3	2	1	0	0	...
	4	...	0	1	3	6	7	6	3	1	0	...
	5	...	1	4	10	16	19	16	10	4	1	...
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

You are to write a function **Plants** that takes two integer arguments, a *column* and a (positive integer) *year*, and returns the number of plants that will be in that column on that year. For example, if (column, year) = (1,5) then **Plants** should return 16. **Plants** should return 0 for any value not in the infinite triangular shaded area.

Plants(column, year)

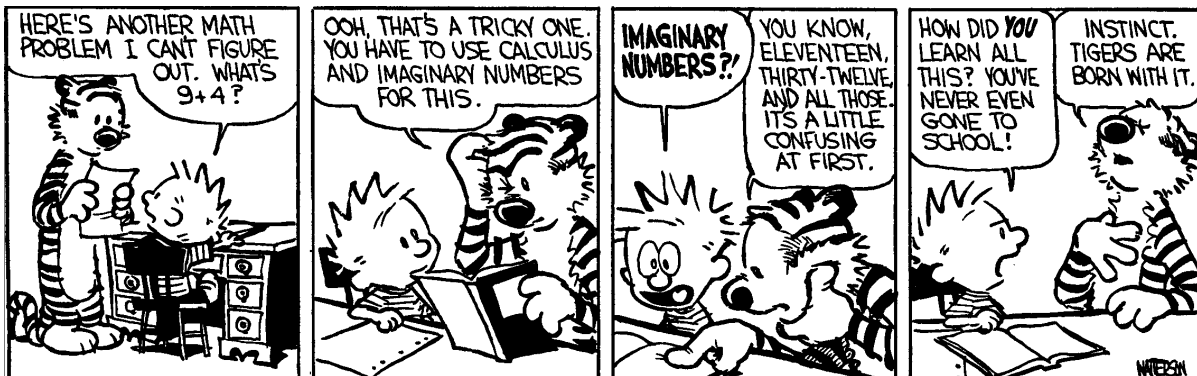
```

if ( _____ )
    report ( _____ )

if ( _____ )
    report ( _____ )

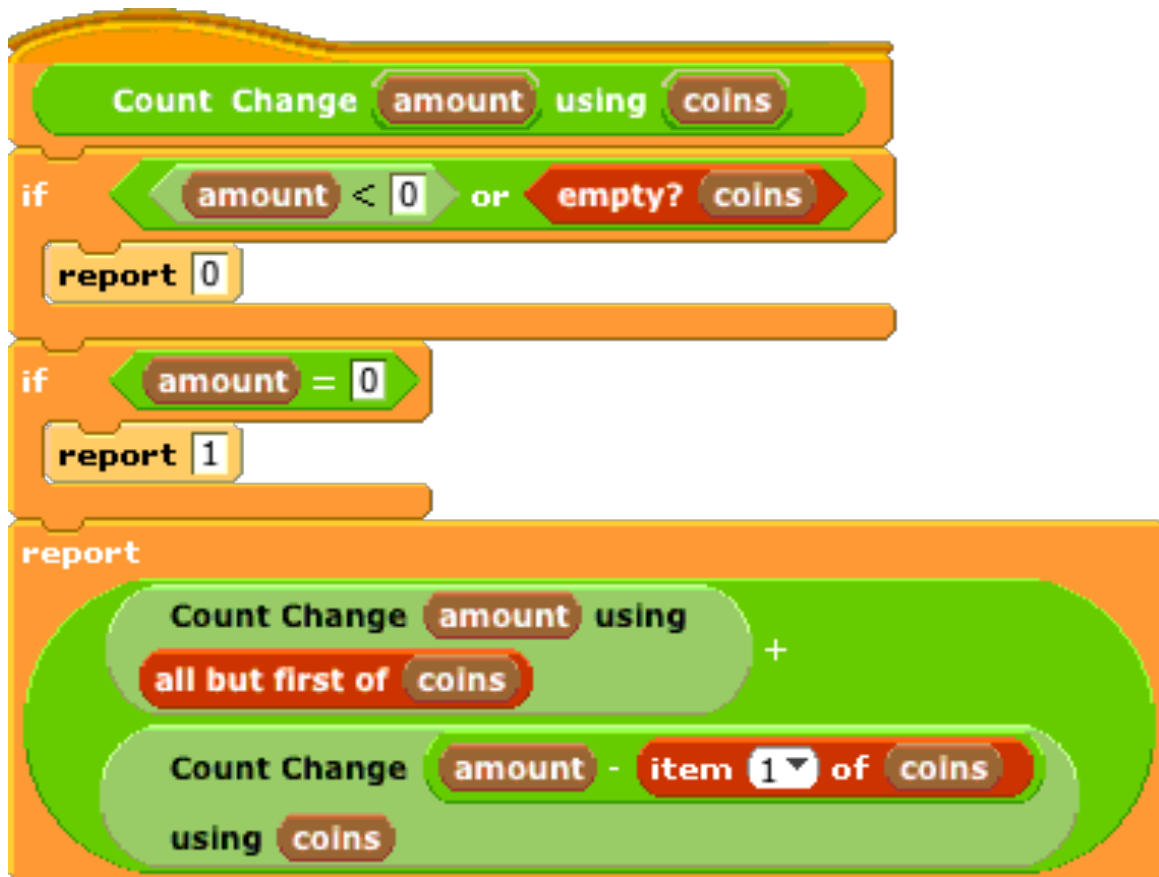
report ( _____
_____ )

```



Question 14: Spare Change? (4 pts)

In lecture we showed **Count-Change**, which takes an amount and a list of coins and returns the number of ways to make change of that amount using those coins. This code is repeated below.



We'd like to see if the code would work if we *reversed* the input list of coins.

On the left is the call tree for the normal sorted-big-to-small coin list (assume a 2¢ coin exists);

draw the call tree for the reversed coin list on the right. Does it work if we reverse the coins? _____

