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

Last Name	
First Name	
Student ID Number	
`cs10-` *Login First Letter*	**a b c d e f g h i j k l m**
`cs10-` *Login Last Letter*	**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**
The name of your LAB TA (please circle)	**Aijia Glenn Luke Navin Rabbit Samir**
Name of the person to your Left	
Name of the person to your Right	
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)***	

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?)

When the right answer would take too long to compute and a close-to-correct answer would be satisfactory (such as weather prediction). It's also possible that optimal answers are desirable (such as AI programs that are designed to play against humans).

+1 in a few cases for statements that are true but not a result of softening the requirement for correctness. One example is when a correct answer doesn't exist.

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?

Energy efficiency (for a fixed computing "load").

+1 for simply stating "efficiency" since there are multiple ways to measure efficiency

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

The redefinition of what knowledge is – if it can't be asked in a multiple-choice exam, it's not knowledge!

+1 in a few cases for minor consequences, but not particularly high-impact ones.

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

That we should move away from standardization (the industrial one-size-fits-all model) That we should support, not suppress collaboration "the stuff of growth"
That we should be thinking differently about human capacity (getting over the old conception of academic/non-academic, abstract/theoretical/vocational)

Additional answers were also accepted. Full or partial credit was given fairly liberally on this problem since it's unclear whether the problem refers to past, current or future paradigms.

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

It allows an author to maintain copyright, and explicitly describe the conditions for sharing, remixing, and reuse of their work without having to be contacted, or involve intermediaries (e.g., lawyers).

+1 if you said that the author retains some rights (or is able to share their content) but don't specify that the author can easily define which rights they want to retain. This was hard to do in the past but CC made it so easy that even a caveman could do it.

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?

The Internet (and mobile phones) will soon connect the entire globe; she's suggesting we think about the ripple effect of our actions, behave as one connected family, and work together to take humanity to the next level.

+1 in rare cases for technologies related to the Internet and mobile phones (but not either directly), or overly general response to the second half of the question.

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?

Researchers revealed that by combining other publicly available datasets, they could identify all of them!

+2 if you mentioned that people were identified using publicly available info.
+1 if you generally mentioned that some people were identified.

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

You no longer had to keep the key secret – you could have Alice and Bob agree on a key without meeting!

+1 if you state something that's true but either (a) occurred at a different time or (b) didn't seriously impact the field of cryptography. The arrival of computers, for example, certainly impacted the field of cryptography but occurred for the first time several decades before (WWII).

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

a9,b3,c7,d2,e10,f4

+1.5 each up to 6 points, then +.5 bonus for each question beyond that.

Login: `cs10-____`

Question 10: *A classy question!* (2 pts)

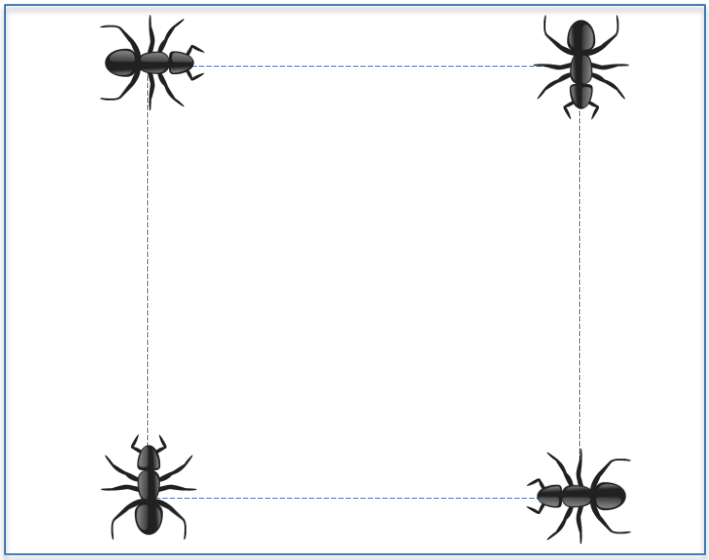
Draw the `CLASSES` list so that the following code returns `cs10`.

The CLASSES list should have another list as its second element, the first item of which is the phrase "CS10"

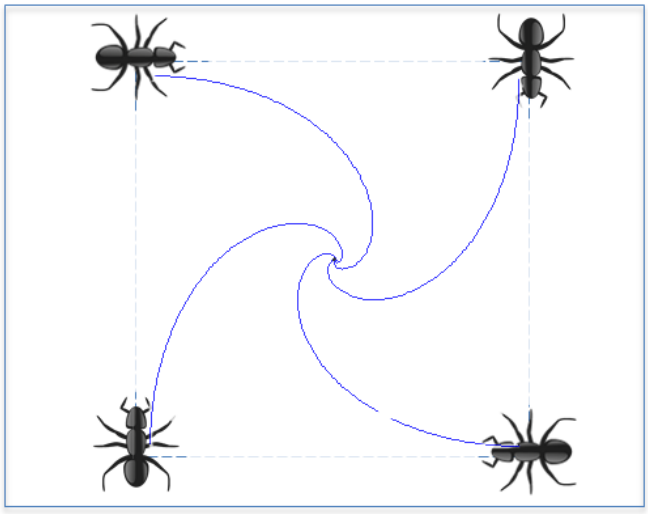
+1 if you swapped the inner and outer lists.

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.



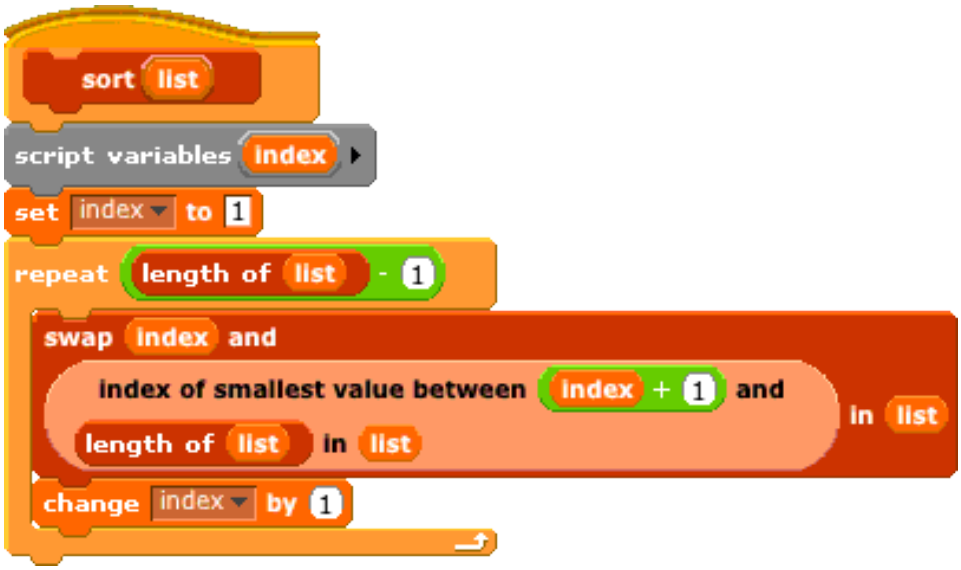
Solution:



2pts for converging at or near center
2pts for some sort of curve towards the center

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
index of smallest value between left index and right index in list	Search the `list` for the smallest value between the `left-index` and the `right-index` (*inclusive*, meaning *including* the elements living at the left and right indices) and report the index of the smallest value in that range
swap left index and right index in list	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`? _____

Quadratic. This is because the "index of …" block must be linear and runs inside of a loop. Both the loop and the "index of …" block are linear, making the whole function quadratic.

+2 for "quadratic"
+1 for anything along the lines of "close to quadratic"

b) If **`list`** is **`(4 3 2 5 1)`**, what is **`list`** after **`sort(list)`**? _____

(1, 2, 4, 3, 5)

+2 for correct list
+1 for (1, 2, x, x, x)

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.

Two obvious patterns emerge: (2 3 4 5 … 98 99 100 1) and (100 1 2 3 … 98 99)

+3 for a correct pattern
+2 for (99 100 1 2 3 ... 96 97 98)
+1 for (100, 99, 98, … , 3, 2, 1)

d) Briefly describe the single, very small change needed to fix the bug.

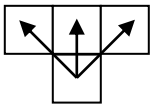
Change index+1 to index.

+3 for mentioning the change from index+1 to index
+2 for fixing bug (only) using a conditional

Login: **cs10-____**

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	C -3	O -2	L -1	U 0	M 1	N 2	3	4	…
	1	…	0	0	0	0	1	0	0	0	0	…
Y	**2**	…	0	0	0	1	1	1	0	0	0	…
E	**3**	…	0	0	1	2	3	2	1	0	0	…
A	**4**	…	0	1	3	6	7	6	3	1	0	…
R	**5**	…	1	4	10	16	19	16	10	4	1	…
	⋮	⋰	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

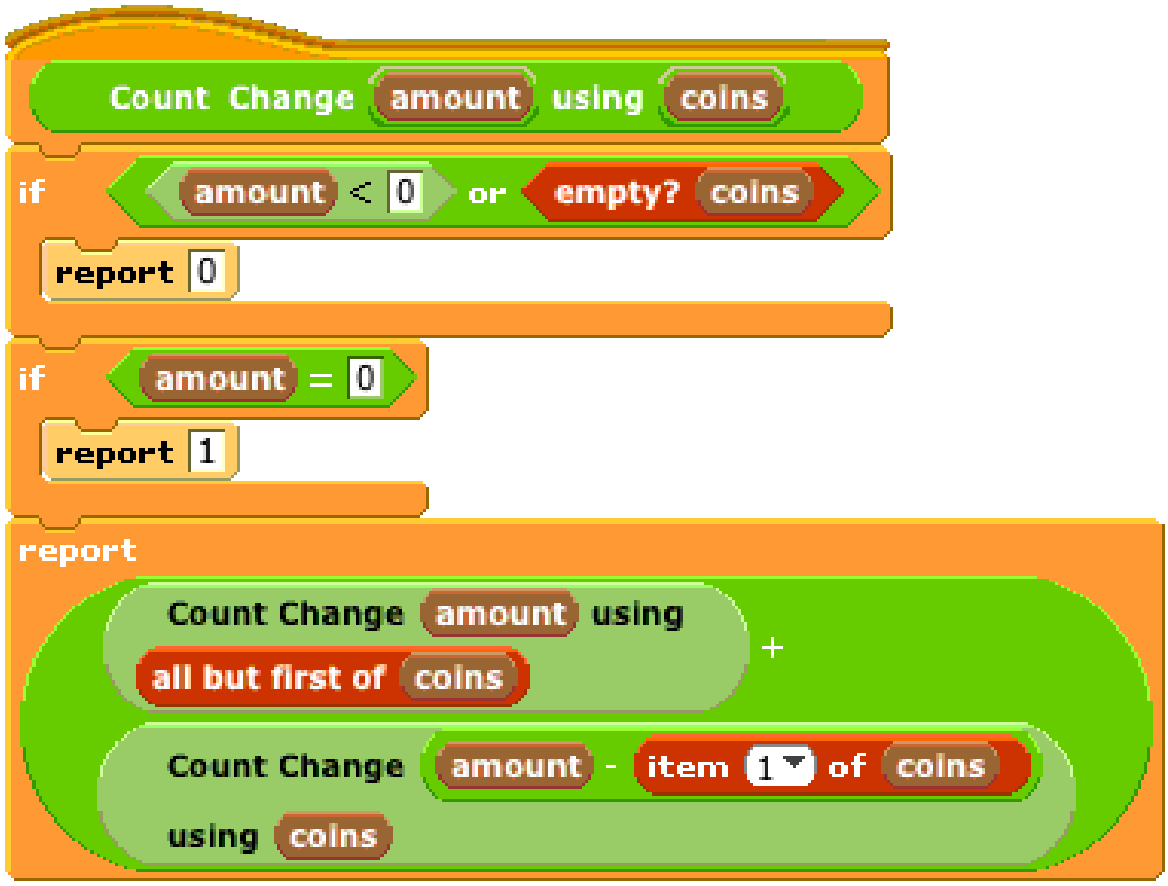
You are to write a function **`Plants`** that takes two integer arguments, a *column* and a (positive integer) *year*, and returns the number of plants that will be in that column on that year. For example, if (column, year) = (1,5) then **`Plants`** should return 16. **`Plants`** should return 0 for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

```
Plants(column, year)
    if ( ___________________ column = 0 and year = 0_____________________________ )

        report ( _______1_______ )

    if ( _________________year = 1_____________________________________________ )

        report ( _______0_______ )

    report (plants(column-1,year-1) + plants(column,year-1) +
            plants(column+1,year-1) )
```

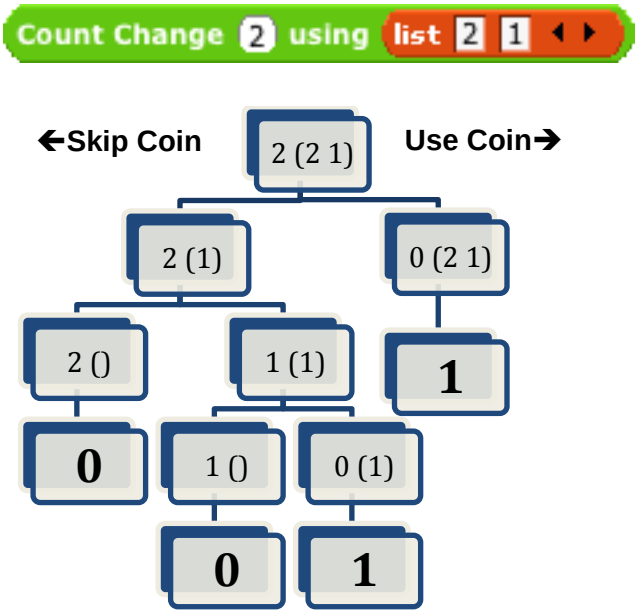
+2 "1" base case test and return value
+2 "0" base case test and return value
+4 recursive case, which should be (+2 three recursive calls and +2 details of arguments)

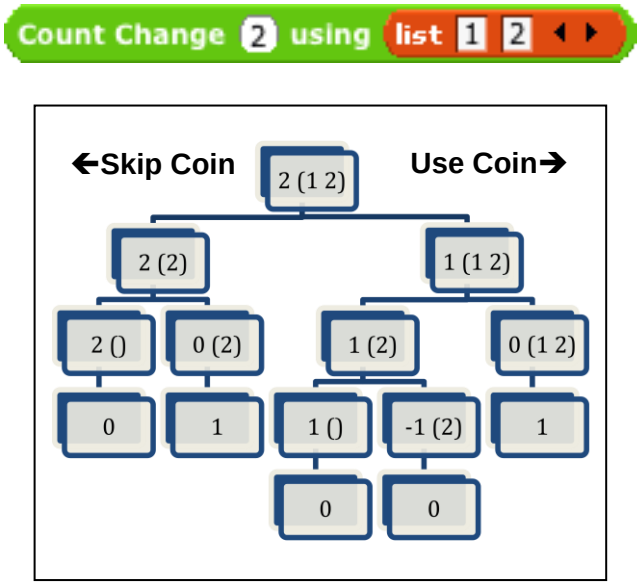
Question 14: *Spare Change?* (4 pts)

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? *yes (1pt)*





3 for correct tree
Partial credit available based on proximity to correct answer

General Grading Rubric for Online Midterms

Each section received a different problem, which is included below. Submissions were all graded according to the same rubric, and the section averages are all going to be curved up to the highest section average to adjust for varying degrees of difficulty.

Out of a possible 10 points:

+2 -- used recursion

+2 -- base case

- +1 stops at the right time
- +1 does the right thing when it stops

+6 -- recursive case

- +2 correct branching factor
- +2 correct move / turns
- +2 on the right conceptual path