University of California, Berkeley – College of Engineering
Department of Electrical Engineering and Computer Sciences
Spring 2013 Instructor: Dan Garcia 2013-05-14

CS10 PAPER FINAL

Last Name	***SOLUTIONS***
First Name	
Student ID Number	
The name of your LAB TA (please circle)	**Aijia Dan Max Michael Pierce**
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 170 minutes to complete this exam. The exam is closed book, no computers, no PDAs, no cell phones, no calculators, but you are allowed **three** double-sided sets of notes. There may be partial credit for incomplete answers; write as much of the single, correct 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**	**15**	**16**	**On-line**	**Sub total**	**1-11 low**	**Total**
Points	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**5**	**5**	**5**	**10**	**10**	**15**	**83**	**3**	**80**
Score	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**5**	**5**	**5**	**10**	**10**	**15**	**83**	**3**	**80**

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

Questions with one sentence answers (3 pts each, lowest dropped)

Question 1: Pandora uses only the Music Genome Project to choose the next song: TRUE / FALSE. Explain.

False, it uses the collective intelligence of BILLIONS of thumbs (up and down) to collaboratively filter choices.

Question 2: Tweets are *by* people *for* people, so companies don't care about Twitter. TRUE / FALSE. Explain.

False, the aggregate "voice" of the people can significantly predict what is hot, and can predict markets.

Question 3: From Anna Rafferty's lecture, how can the field of AI help with education? *(what she works on)*

By modeling human learning using machine learning, and providing adaptive feedback and instruction in computer-based environments. E.g., diagnosing a student's knowledge by watching her play a game.

Question 4: Aside from what Google and Twitter do, what is one useful application of *Big Data*?

Identifying signatures for diseases, adapting lights/navigation/tolls based on Traffic, Natural disaster prediction

Question 5: Why is MapReduce a great distributed computing abstraction?

It hides the details of load balancing, failed machines, and data distribution from the programmer.

Question 6: The *subset problem* (given a list of numbers, find some subset of them that adds to 0) is:

Tractable Intractable-with-no-approximating-algorithm Intractable-with-an-approximating-algorithm undecideable

Intractable-with-no-approximating-algorithm

Question 7: What is this?

What humans look like to a computer today (finger to type, ear to hear sounds, eye to see screen)

Question 8: From *Blown to Bits:* who are the *intermediaries* of the Internet when data is sent from person-person?

Internet Service Providers (and also the web service, like MySpace, FaceBook)

Question 9: What was the specific challenge for the robots in "The Great Robot Race"?

To drive, unmanned, 130 miles across the Mojave desert.

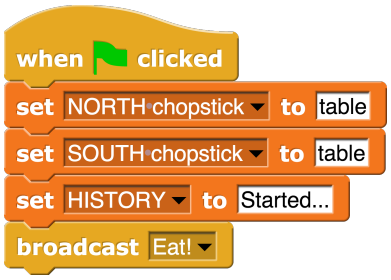
Question 10: What was one thing Alan Turing did that fundamentally changed computing (or the world)?

Broke the German code in WWII, Asked early questions about what it meant for a machine to be intelligent, Defined a notion of equivalence of power of languages, through the notion of a universal machine. Simulation

Question 11: The two pillars of science have always been *theory* and *experimentation*. What's the third?

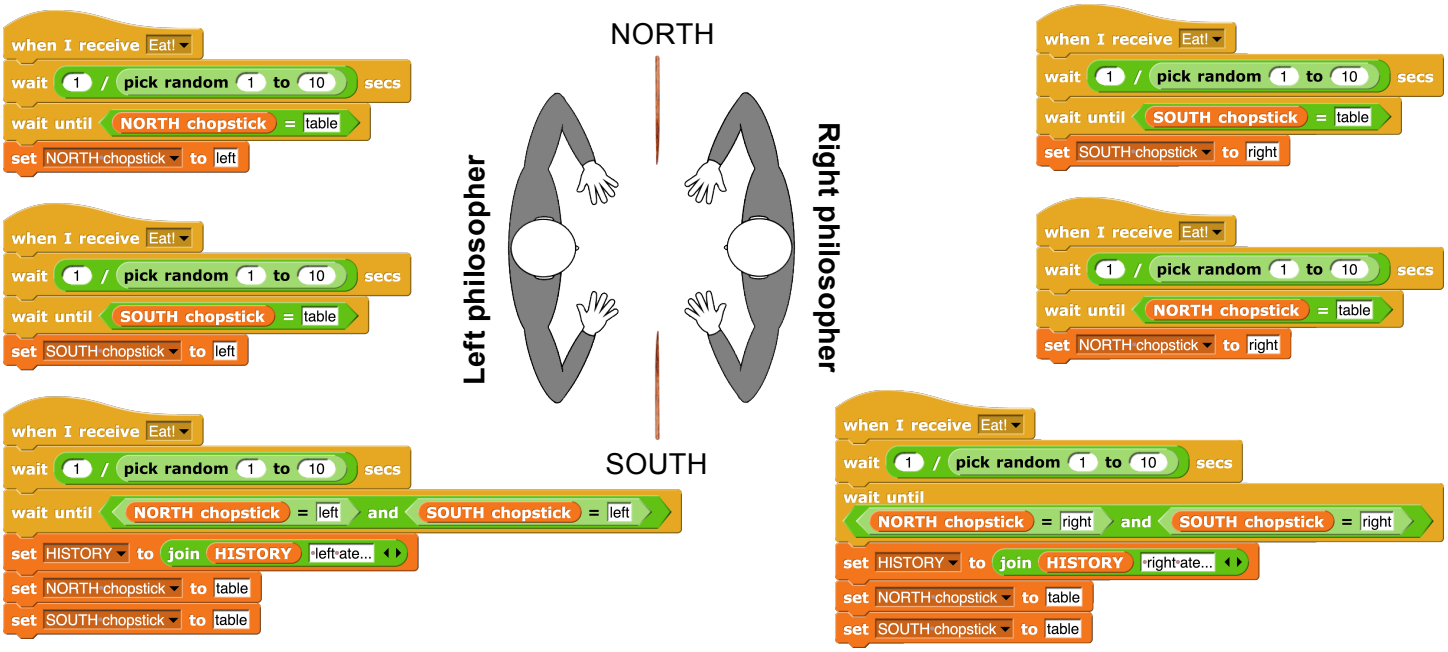
SID: _____

Question 12: *Dining Philosophers* (5 pts)



Two philosophers (left and right) are having dinner, sitting across from each other. There is a NORTH and a SOUTH chopstick on the table. Each philosopher continually looks down to see if a chopstick is on the table, and tries to grab it; if both are ever grabbed by one person, that person eats, updates HISTORY (a record of what happened) and puts the chopsticks down. Ten seconds after the green flag is clicked, what could HISTORY be?
(all the boxes are not necessarily needed)

Started… ***Left ate…*** ***Right ate…***	***Started…*** ***Right ate…*** ***Left ate…***	***Started…***		



Question 13: *Beethoven wasn't the only great composer…* (5 pts)

We've provided some helper reporter blocks that work on numbers. Use the techniques from "Writing Scratch/BYOB code on paper". E.g., would be written **`increment(square(4))`** ➔ **`17`**.

Block	Description	Example
`square`	Reports the square of the input	square 10 → 100
`increment`	Reports the number one larger than its input	increment 10 → 11
`reverse`	Reports the number with its digits reversed. All leading zeros in the reversed number will be removed.	reverse 120 → 21

4

a) What input causes the desired output? **`reverse(square(increment(_____))`** ➔ `52`

b) Fill in the blank on the left (with calls to *at most 5* of these reporter blocks: `square`, `increment`, and `reverse`) so that the expression reports 9. *The blank on the right is for matching right parentheses.*

square (increment (reverse (increment (reverse ***)))))***

_____ `910` _____ ➔ `9`

Question 14: *I still like to* `Draw Pictures`... (5 pts)

Given the definition below, *draw the resulting pen marks* of 

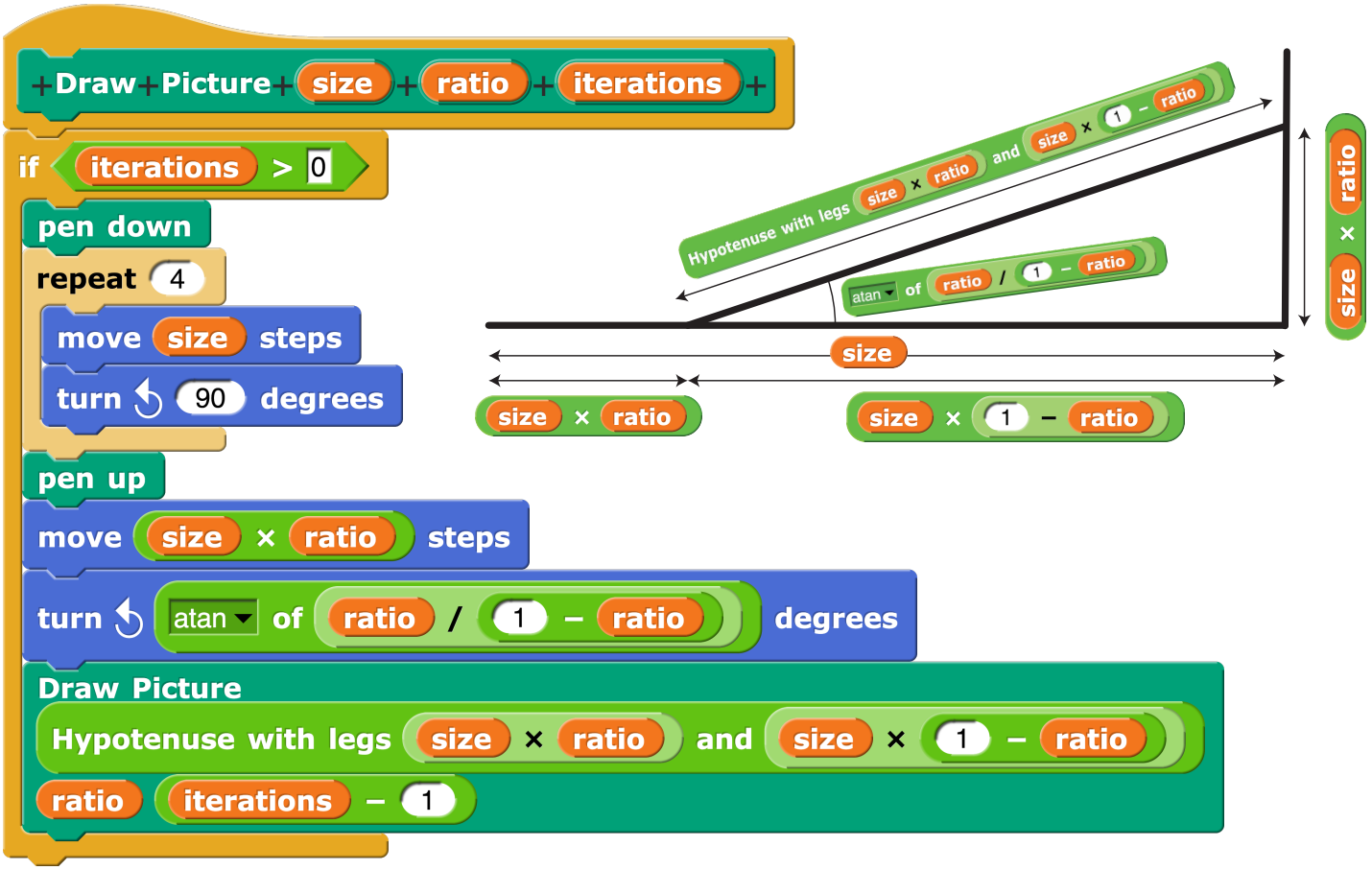
.

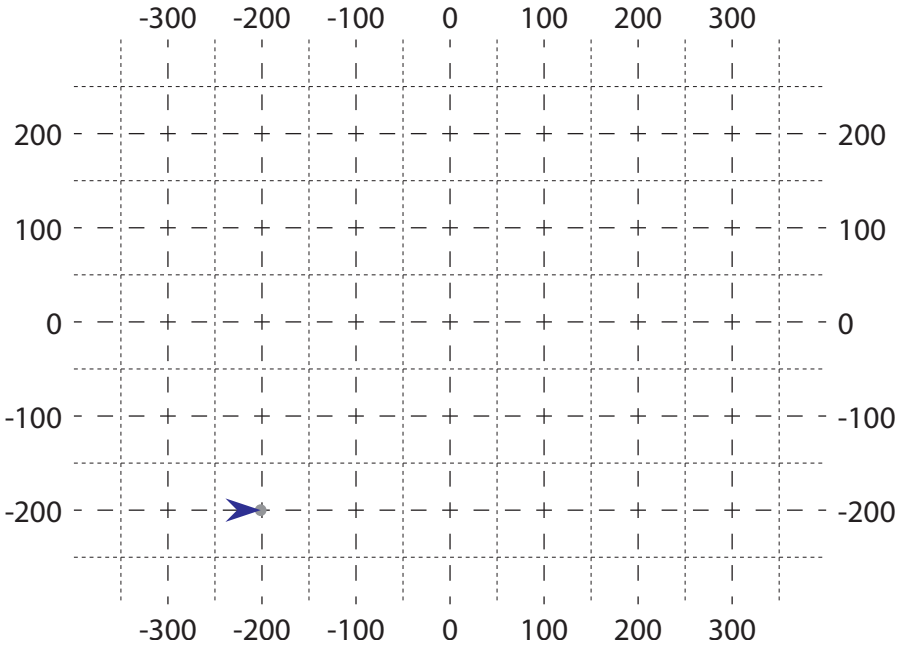
The diagram below is intended to explain the math that is used in the block definition.

The "arrowhead" sprite starts positioned at (-200,-200) pointing right, and its pen is at its tip, and *down*.

Assume the sprite can draw anything between -400 to 400 horizontally and -300 to 300 vertically.

You do NOT have to do any math for this problem … use the diagram below and approximate.

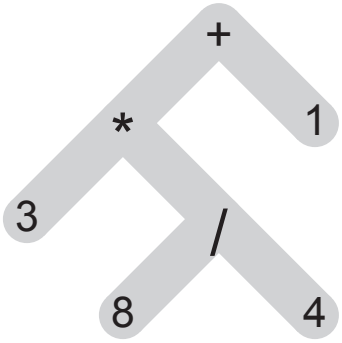
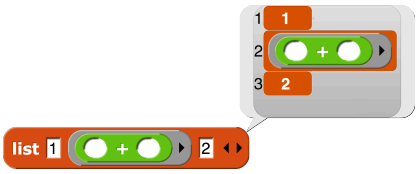


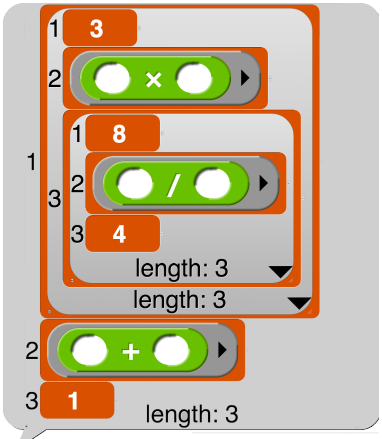


SID: _____

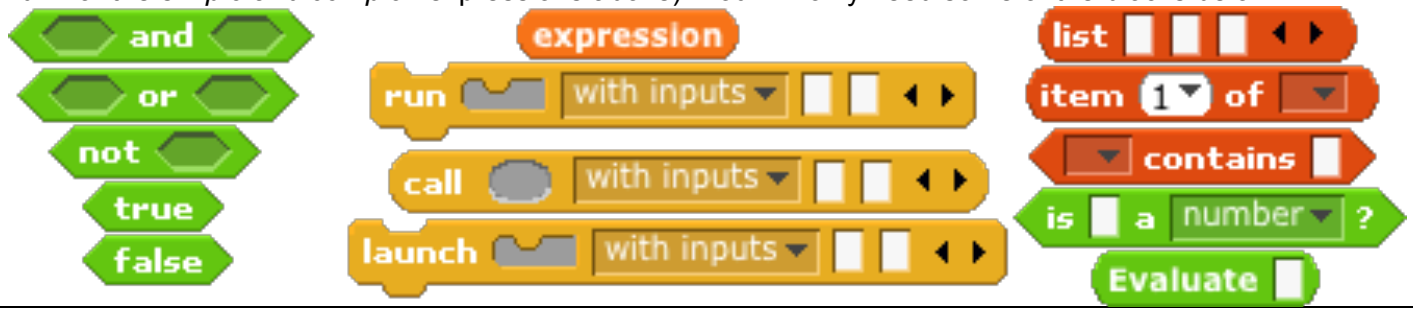
Question 15: *You're such an expressionist…* (10 pts)

One of the things you've learned in CS10 is that, thanks to abstraction, there are often multiple ways to represent the same thing. Consider mathematical expressions; they can be represented in a **WRITTEN** way, in a **GRAPHICAL** way, and in a **LIST** way. Here are two examples for a *simple* and *complex* expression:

	Simple (value = 3)	*Complex* (value = 7)
WRITTEN	1 + 2	(3 * (8 / 4)) + 1
GRAPHICAL		
LIST		*(answer won't fit in this box… see the list immediately below and to the right of this box)*



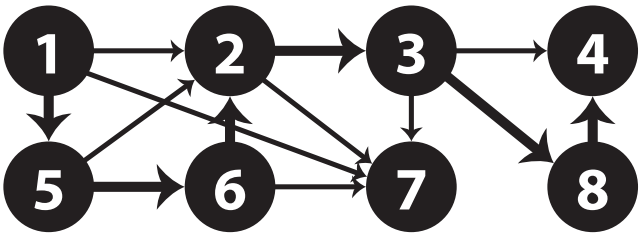
Write **`Evaluate`** that should take the expression as a **LIST** *(or number)*, compute and return the value (here, 3 and 7 for the *simple* and *complex* expressions above). You will only need some of the blocks below.



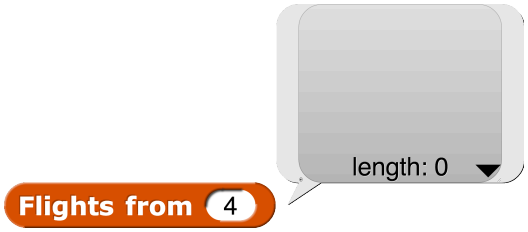

```
Evaluate(expression)
                    is(expression)a-number?
    if ( _____ )
                    expression
        report ( _____ )
    else
                    call(item(2)of(expression))with-inputs
        report ( _____
                    Evaluate(item(1)of(expression))
                _____
                    Evaluate(item(3)of(expression))
                _____ )
```

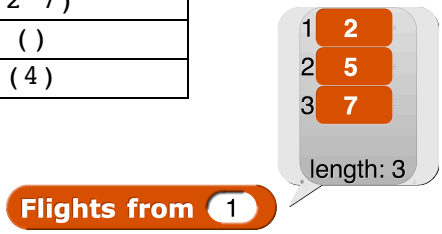
Question 16: *I love to visit cities…* (10 pts)

It's the day after the entire country has descended on a particular city and it's now time to fly them all home. Airlines have rearranged their flights so that all flights are *away* from the original city (i.e., there are **no loops** that might allow a passenger to start at one city, take some flights, and return to that city). Each of the cities are numbered *1* to CITIES; cities connected via a direct flight are reported via Flights from city. E.g.,

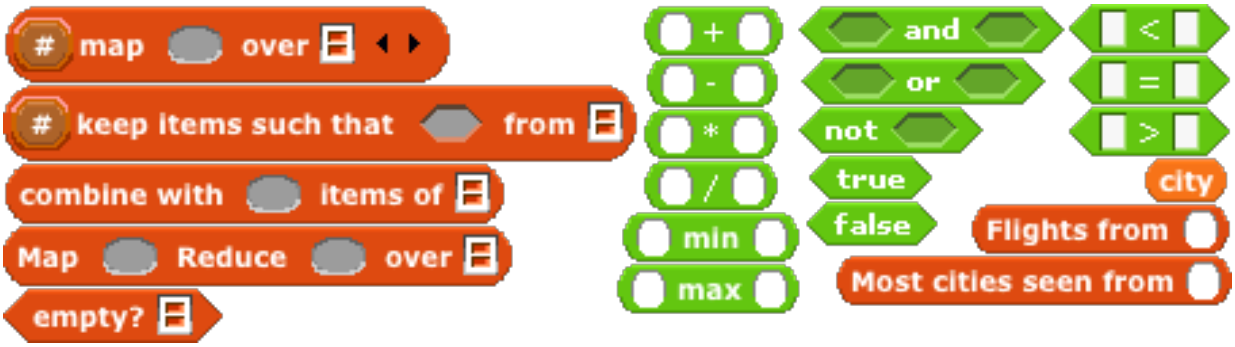


city	Flights from city
1	(2 5 7)
2	(3 7)
3	(4 7 8)
4	( )
5	(2 6)
6	(2 7)
7	( )
8	(4)





A passenger comes up to the airline desk and says she loves to visit cities, and wants to know what is the **most cities she could visit** (*including* the departure and destination city). We've highlighted the route in **bold** above that she would take, which would allow her to visit 7 cities if she started from 1: (1, 5, 6, 2, 3, 8 and 4). Your job is to write this block that should work for any graph. The nice thing is you'll only need the following blocks below. **( )min( )** and **( )max( )** return the smaller and larger of their two inputs, respectively.




```
Most-cities-seen-from(city)
                    empty?(Flights-from(city))
    if ( ____________________________________________________________ )
                    1
        report ( ____________________________________________________ )
    else
                1 + Map[Most-Cities-Seen-From]Reduce[max]over(Flights-from(city))
        report ( ____________________________________________________
                 ____________________________________________________ )
```