

University of California, Berkeley – College of Engineering

Department of Electrical Engineering and Computer Sciences

Spring 2012 Instructor: Dan Garcia

2012-03-22

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 Luke Navin Pierce 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	2	2	6	4	10	10	10	60
Score	2	2	2	2	2	2	2	2	2	2	6	4	10	10	10	60

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

Questions with one sentence answers (2 pts each)

Rubric: 2 pts perfect, 1 pt close, 0 pts not close

Question 1: One big idea in this course is that often technology has unintended *negative consequences*. What was one of the negative consequences of the advent of geo-tagged and date-stamped YouTube videos?
Safety issues, privacy issues, etc. Geo-tagged and data-stamped videos tell the public where you are, who you are with, what you are doing, and using that info, criminals can break into your house. (People could cybercase you and find out when you were on vacation and rob your house when you're away.)

Question 2: Amdahl's law states that if a program contains a 1/5 serial (non-parallel) portion and 4/5 parallel portion is run on a fast machine with a near-infinite number of cores, what's the *most speedup you could ever achieve* over a machine with a single core?

Speedup $\leq 1/s = 1/(1/5) = 5x$

Question 3: Judah Schwartz said there was a continuum of computer uses in education: from *tools* through *microworlds* through *courseware*. BYOB would be a tool, but what would a microworld for teaching graphically recursive structures in BYOB look like?

It would be a fractal exploration tool, allowing you to easily click and place and move lines and shapes and curves, and it would render the fractal at different recursion levels instantaneously...no programming needed.

Question 4: One of the most important aspects of HCI is the *Iterative Design Cycle*. What is that?

Design → Prototype → Evaluate → (repeat)

Question 5: What is the “free performance lunch” that’s now over?

For years, software applications got faster, without developers needing to do anything, because the underlying hardware (CPU, memory, disk) was faster.

Question 6: Computer performance is continuing to increase exponentially. This means one by one we’ll be able to *strongly solve all the world’s board games*. True or False? Explain.

False. The game Go has 3^{361} positions (10^{170}), far more than the number of atoms in the universe (10^{80}).

Question 7: From an engineering perspective, Twitter doesn’t want everyone with an account to “follow” everyone else with an account. Why would the financial officers Twitter not want that either? (*Hint: think about what Rushkoff said in his “Program or be Programmed” video – what’s really behind Twitter’s financial model?*)
If everyone follows each other, Twitter won’t be able to identify “who-likes-what” or “who-follows-who” pattern, and Twitter will lose commercial benefits because they can’t make money through ads based on what people like. Also, if everyone follows each other, people would become equally popular, and there would be no incentive for people to be on Twitter.

Question 8: What is one reason (also provide an example) most of us have *given away* our privacy?

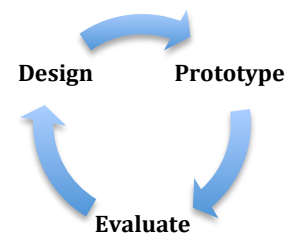
Saving time (e.g., toll booth transponder), saving money (e.g., supermarket loyalty cards), convenience (e.g., Amazon providing suggestions to us), or we can’t live any other way (e.g., credit cards) Or we like it!

Question 9: If the US Government wanted to perform mind control, what could they require Google to do?

The government can ask Google to modify or even take out the search results they don’t want people to see.

Question 10: Recently, there was “de-escalation” in the age-old copyright war; describe it.

Recording labels now regularly offer music DRM-free; they’d said they’d never do it. Also Creative commons



Question 11: *Beethoven wasn't the only great composer...* (6 pts)

We've provided some helper reporter blocks that work on *both* words and sentences.

Block	Description	Word example	Sentence example
Length	Report the number of letters in a word / words in a sentence		
Unend	Remove the ends of a word / sentence.		
Double	Double a word / sentence		
RightDup	Duplicate the rightmost letter / word		
LeftTrim	Remove the first letter / word from the left		

Fill in the blanks below with only calls to the reporter blocks above: Length, Unend, Double, RightDup and LeftTrim so that the expressions evaluate correctly. Use the techniques from "Writing Scratch/BYOB code on paper". E.g., would be written `RightDup(Double(Bears))`.

You may use at most 5 reporter blocks in any one blank. The blanks on the right are for matching right parens.

(Length (RightDup (Double (Unend

a) _____ (I love cal)___ → 9

(LeftTrim (Unend (Double (Unend (Unend

b) _____ (go bears and beat stanford)___ → dan

These were each worth 3 points. Any combo resulting in the correct output was worth 3 points, however, if more than 5 blocks were used, 1 point was subtracted. If the answer was very close to being right (eg "dand"; switching two blocks; having all the blocks in the reverse order, leaving off a single block like length) then 1 point was removed. A single point could be awarded in the case where the student had the correct method but did not finish proper implementation.

Question 12: Three body simulation... (4 pts)

You have three sprites, A, B and C. A is centered on the screen and does nothing (never moves). B and C follow the scripts below, and the pen for C is in the center of the diamond. Draw the pen marks that are made. Assume sprites are allowed to go off the stage.

Solution: B orbits (goes in a circle around) A, not leaving any pen marks. At the same time C starts to orbit around B. This creates a Sun-Earth-Moon relationship. C leaves a spiraling circle behind it. (as shown below)

4 points - Any kind of spiral (1 is enough, because of the chance that C will be catapulted out of orbit)

3 points - Minor errors, but on the right track.

2 points - Concentric circles or similar thought tracks.

1 point - Showing that C curved.

The image shows the Scratch code for two sprites, B and C, and a diagram of their initial positions.

Sprite B Code:

- when green flag clicked
- forever loop:
 - point towards A
 - turn 90 degrees
 - move 1 steps

Sprite C Code:

- when green flag clicked
- clear
- pen down
- forever loop:
 - point towards B
 - turn 90 degrees
 - move 3 steps

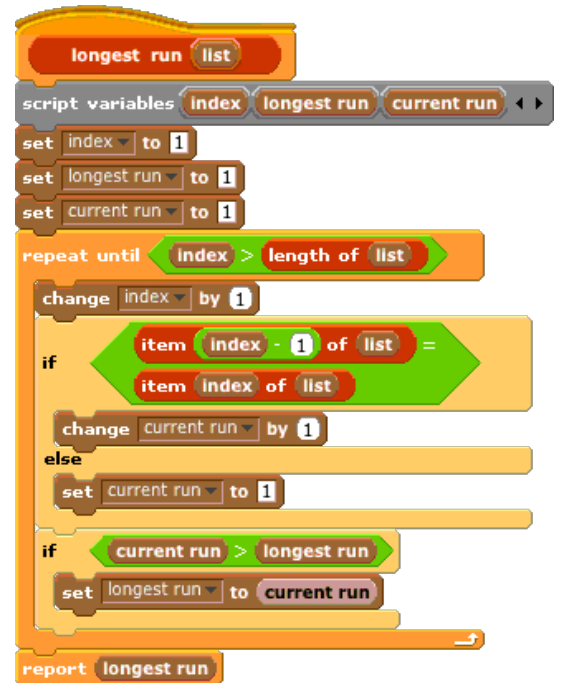
Diagram: A rectangular box shows the initial positions of the three sprites: A (a circle), B (a triangle), and C (a diamond). A is on the left, B is in the middle, and C is on the right.

The stage shows the path of sprite C. A yellow mouse cursor is at the center of the stage. A blue triangle (B) is at the bottom right. A blue diamond (C) is at the bottom right, with a blue pen mark at its center. The path of C is a spiraling circle that starts at the center and moves outwards, crossing itself multiple times.

Login: cs10-_____

Question 13: *I've got this incredible run going, hold on!!* (10 pts)

When a list has many consecutive identical elements in a row, it is called a “run”. We’d like to write a reporter, `longest-run(list)` to return the maximum number of consecutive elements in a list. For example, `longest-run((a b c d d e a b x x x d q))` would return 3 because the highest number of consecutive elements is 3 (the run of 3 x’s in a row). Assume lists have at least one element in them. We’ve tried to write the code ourselves, but unfortunately, it has a bug.



```
longest-run list
script variables index longest-run current-run
set index to 1
set longest-run to 1
set current-run to 1
repeat-until index > length-of list
  change index by 1
  if item index - 1 of list = item index of list
    change current-run by 1
  else
    set current-run to 1
  if current-run > longest-run
    set longest-run to current-run
report longest-run
```

a) If `length-of(list)` is a linear operation, what is the order of growth of `longest-run` quadratic?

This question was worth 2 points and no partial credit was awarded for incorrect answers.

This was worth 2 points. The grading was 2 points for a correct answer and none for a wrong answer. The reason the answer is quadratic is because length of list is a linear operation which is run each time the repeat until loop is called. Therefore the operation is length of list times length of list times. Quadratic.

b) Complete the sentence. The shortest list that triggers the bug is aabb.

This question was worth two points and the answer required giving a list which correctly revealed the bug. (Any list of the form XXYY was accepted for full points.) We awarded 1 point partial credit for lists that mostly understood the bug, but that was too long (like aabbb, or aaabb). We also marked off 1 point if the answer was 4 but didn’t actually specify what the list looked like. Most other answers received no credit if the list was too long or didn’t show any understanding of the bug.

The `longest-run` function *should* return 2 for this list, but our buggy version returns 3.

This question was worth 2 points and we gave partial credit if the difference between the two errors was 1 (worth 1 point) or if the errors were correct but given in the wrong order. All other answers received no credit.

Describe the smallest fix that fixes the function so that it works as advertised on all non-empty lists.

This was worth 4 points and the simplest was to change the first “if” block to an “if-else” block. We awarded partial credit for many varieties of answers which showed some understanding of what the correct error was. In some cases students didn’t correctly identify the error in part B and while their solution may have fixed that error it received no credit.

Change the first if to an if...else, with the else clause containing “set current run to 1” (as shown above).

Question 14: Was this game invented in Italy in the 13th century? (10 pts)

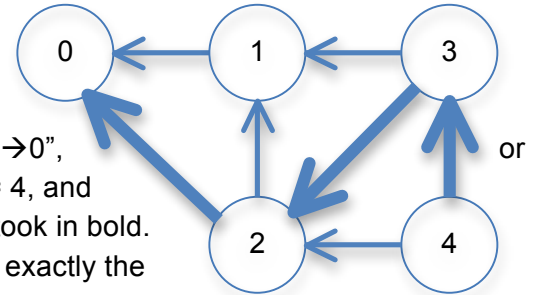
Consider a game similar to the “1,2,...,10” game from lecture. In this game, two players are given a bowl of N candies on the table. They take turns, eating ONE or TWO candies from the bowl. The first player to eat the last candy and leave the bowl empty is the winner. Here’s an example game:

Ana: “Hmm, the number of candies in the bowl is 4. I’ll eat ONE to make it 3.”

Bob: “I’ll eat ONE to make it 2.”

Ana: “I’ll eat TWO to leave an empty bowl. I win!!”

You want to calculate how many different paths (i.e., different games) there could be starting with N candies. For example, with 4 candies, one path was the one Ana and Bob played: “4→3→2→0”. Another might have been “4→2→0”, or “4→3→2→1→0”, or “4→2→1→0”, “4→3→1→0” (5 paths in total). You decide to visualize this when $N = 4$, and you get the picture on the right. We’ve shown the path Ana and Bob took in bold. You realize the number of different paths in a game with N candies is exactly the same as the number of different ways to go from N to 0 in a visualization like the one shown here.



a) Write a function **Paths** that takes one integer argument N (the starting number of candies), and returns the number of paths in the candy game starting with N candies. For example, **Paths(0)** should return 1 (there’s only one path if you’re already there: “0”). **Paths(1)** should return 1 (one path: “1→0”). **Paths(2)** should return 2 (for the two paths: “2→0” and “2→1→0”), **Paths(4)** should return 5 (as described above), etc.

Paths(N)

```
if ( <N = 0> or <N = 1> OR <N < 2> )
```

This was worth 2 points and the most common errors were forgetting to account for both Zero and One base cases (minus one point) or having a recursive call to Paths() in the base case. This would cause an infinite loop and was minus 1 point. Other variations were awarded 1 point if the showed understanding of the way the base case worked but made a critical error.

```
report ( 1 ) This was worth 1 point and there is no partial credit.
```

```
report ( Paths(N-1) + Paths(N-2) )
```

This part was worth 4 points and the most common errors were not actually making the recursive calls recursive (not calling Paths) which was minus two points, or forgetting one of the recursive calls (like Paths(N-1)) which was also minus two points. Some other common mistakes were not separating the recursive calls correctly which was worth minus 3 points. Other solutions were awarded partial credit based on the level of understanding of the recursive paradigm.

b) What is the order of growth of **Paths(N)**? Exponential or $O(2^n)$

This question was worth two points and there was no partial credit.

c) The values returned by **Paths(1)**, **Paths(2)**, **Paths(3)**, etc.

form an interesting series. Who was the first person to notice this? Fibonacci

This was worth 1 point and no partial credit was awarded.