CS10 Fall 2017 Final Exam Answers

Question 1: Which were not discussed in the "Saving the world with computing" lecture?
While some projects involve teams, the majority of the projects were completed by individual researchers.

Question 2: Which of the following is *not* usually characterized by the time "after the singularity"? Moore's law ends

Question 3: When we solved the "10…0" game in class (the game starts with 10 sticks, each player removes 1 or 2, first person to get to 0 wins), it worked fine. However, when we tried to solve the "50…0" game (same rules as the original game, except we start from 50 instead of 10), it didn't return even though we waited a long time. Why would that be? We were doing lots of redundant calculations. It's exactly like our slow, recursive Fibonacci (the trees look almost identical).

Question 4: It would be nice to know if our program from question 3 above would *ever* (or *never*) return. What should you say if someone wanted to write a program that would take in a program (e.g., **solve**) and input (e.g., **50**) and return whether that program run on that input will *eventually* return? *It's not possible to write that program! So proved by…*

Question 5: who? Alan Turing, who showed that this task, otherwise known as the Halting Problem, was undecideable.

Question 6: What is (111_3 + 11_2) written in Hex? (9+3+1) + (2+1) = 16 = 0x10 = 10_{16}

Question 7a: What input to the nested blocks `swap(right( ))` would cause them to output RATS? RTSA

Question 7b: What set of nested blocks if called on STOP would return SPOT?

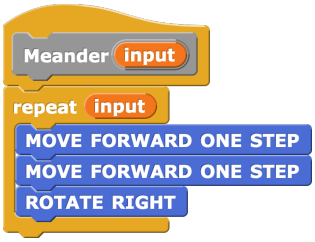
Question 8: Write ALL the possible values of **SALARY**. Left (serial): 14; Right (parallel): 11, 13, 14 (code below)

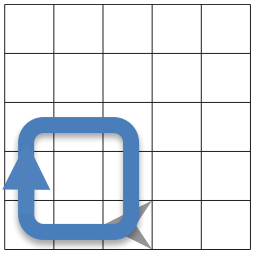
Question 9a: What is running time of `sort`? Quadratic (the for loop is linear, and each step involves index, also linear).

Question 9b: If data is (4 3 2 5 1), (i.e., a 5-element list whose first element is 4, second element is 3, etc. and whose last element is 1), what is data after sort(data) runs? (43251) → (13254) → (12354) → (12453) → (12435). `sort` code below.

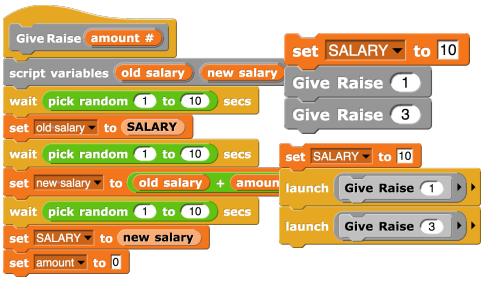
Question 9c: What 10-element list (containing the numbers 1 through 10 in some order), when passed to `sort`, would be correctly sorted after `sort` runs? (10 1 2 3 … 7 8 9), (2 3 … 8 9 10 1)

Question 9d: Briefly describe the single, very small change needed to fix the bug. Decrease δ by 1 (so the innermost call becomes: "index of smallest value between (index) and (length of (data))")





```
def dictionary_reverser(D):
  R = {}
  for k in D: # keys
    if D[k] in R:
      R[D[k]].append(k)
    else:
      R[D[k]] = [k]
  return R
```



Question 10a: Which shows a possible result of calling **Meander**? (it turns right and makes a square 3 boxes on a side)

Question 10b: If the input to **Meander** were ∞, how many *total different squares would ever be visited*? 8 (shown)

Question 11a: *Which of the following could be answered by analyzing only information in the data set?:* How many states have a higher % of female computer science majors than male computer science majors attending college in that state?

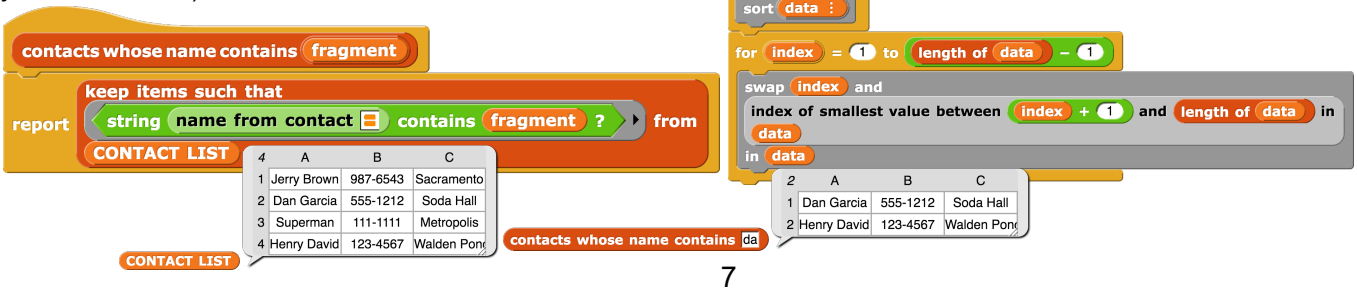
Question 11b: *How could we extract a list of the states (with no duplicates) with CS majors from the data set in the fewest steps?* First, use **map**, and then remove duplicates (map across the data set to extract the state from each student record, which gives us a list of all the student's states, and then remove duplicates to get just the states)

Question 12a: What does `["cal", "berkeley", "stanford"][1][2]` evaluate to? `"r"`

Question 12b: What does `[x*10 for x in range(3) if x != 1]` evaluate to? `[0, 20]`

Question 12c-h: We want to write a dictionary reverser. See above in the text box

Question 13: Block that takes in a word fragment (the kind you'd type to a search prompt) and returns the *contacts* (not just the names!!) from a contact list whose names match.



	A	B	C
1	Jerry Brown	987-6543	Sacramento
2	Dan Garcia	555-1212	Soda Hall
3	Superman	111-1111	Metropolis
4	Henry David	123-4567	Walden Pon

	A	B	C
1	Dan Garcia	555-1212	Soda Hall
2	Henry David	123-4567	Walden Pon