CS10 Fall 2016 Final Exam Answers

Question 1: Who gave the "Mother of all Demos", introducing the mouse to the world? … Doug Englebart

Question 2: The internet uses an *end-to-end architecture*. That means… None of the above. It means that all the ENDPOINTS have all the "intelligence", the opposite of option 2.

Question 3: A polynomial-time solution to the knapsack problem means that for the **first time**…We can now solve the *subset sum* problem in polynomial time, since the knapsack problem is NP-complete, and the subset sum can be reduced to the knapsack problem in polynomial time.

Question 4: Which of the following is **true**? *Fortran* is the most common programming language in scientific code.

Question 5: Which of the following are part of the traditional "interface design cycle"? Design, Prototype and Evaluate.

Question 6: Why did we move from IPv4 to IPv6? Because IPv4… didn't have enough addresses.

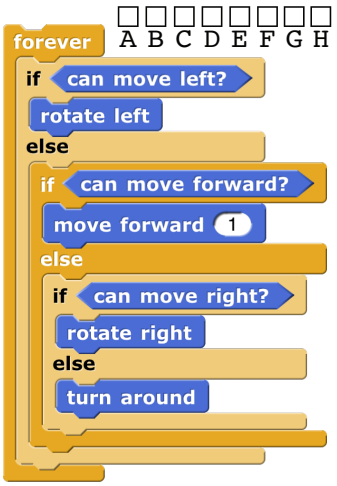
Question 7: What did the halting problem prove? Not all problems were *decidable.*

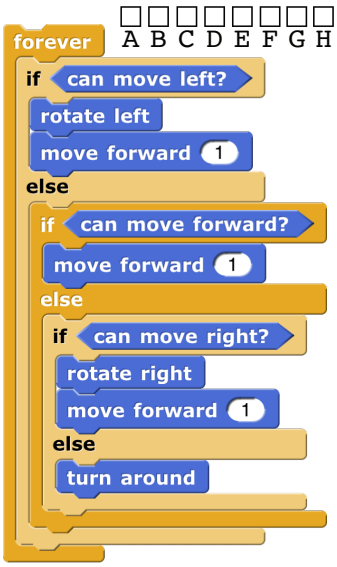
Question 8: What is the "Attack of the Killer Cellphones"? Parallel system architects are looking to cell phone processors to understand how to manage power better.

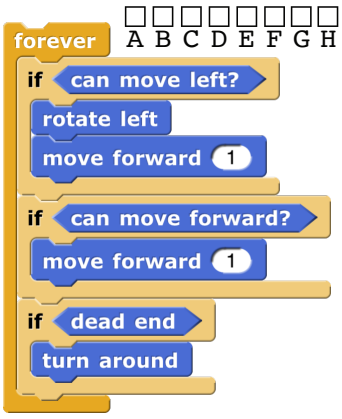
Question 9: You plan to test an algorithm with a set of very extensive test cases. Which is true? None of the above.

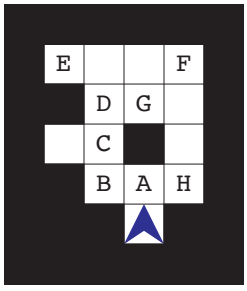
Question 10: YouTube now uses 64 instead of 32 bits to count views. How many more is that? 2^{32}x

Question 11: A coin has two sides, labeled "1" and "2". Consider the goal of simulating the results of flipping the coin *five* times, and displaying the *sum* from the five flips. Which of the following code segments will produce the appropriate results? *Hint: Compare the # of ways there are of summing to 5 vs. 7…* None of the above, the only way to do it would be to add up the result of five calls to "`pick-random(1)to(2)`".









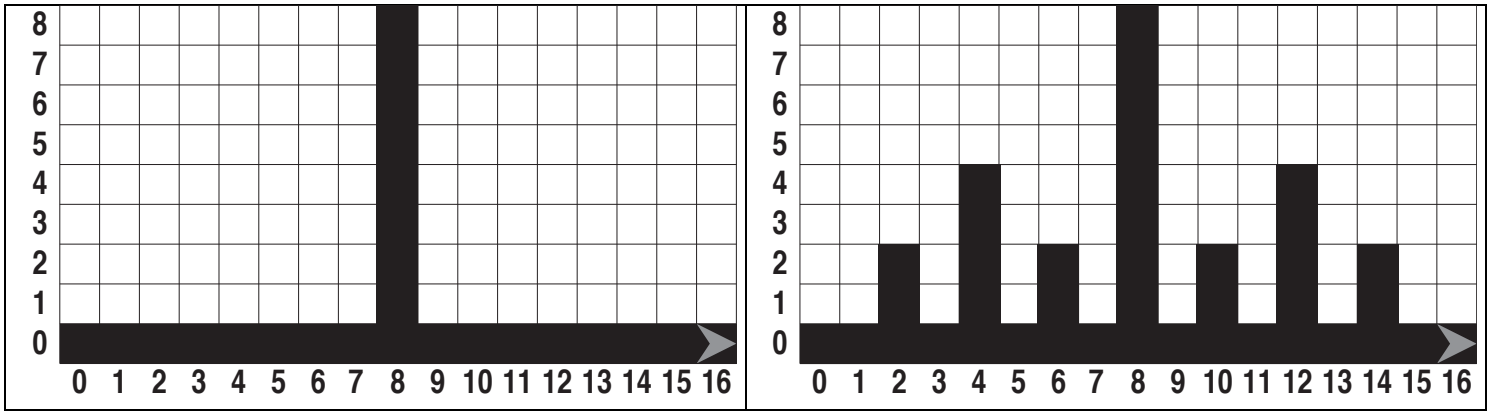
Question 12a: AH (Iterations: (1) Can't move left, moves up to A(facing north).
(2) Can move left to A(facing west). (3) Can move left to A (facing south).
(4) Can move left to A (facing east). (5) Can't move left, can move forward to H (facing east).
(6) Can move left to H (facing north). (7) Can move left to H (facing west).
(8) Can't move left, can move forward to A (facing west). Go to Iteration 3 which causes it to visit A,H forever.)

Question 12b: ABCDEFH (walks around the "left wall" of the maze forever, missing "G")

Question 12c: ABCDEFH (same as 12b, because it makes right turns by turning around and then turning left!)

Question 12d: AB (Iterations: (1) Can't move left, moves up to A(facing north), not at a dead end. (2) Can move left to B(facing west), can't move forward, not at a dead end. (3-∞) Can't move left, can't move forward, not at a dead end.)

Question 13a,b:



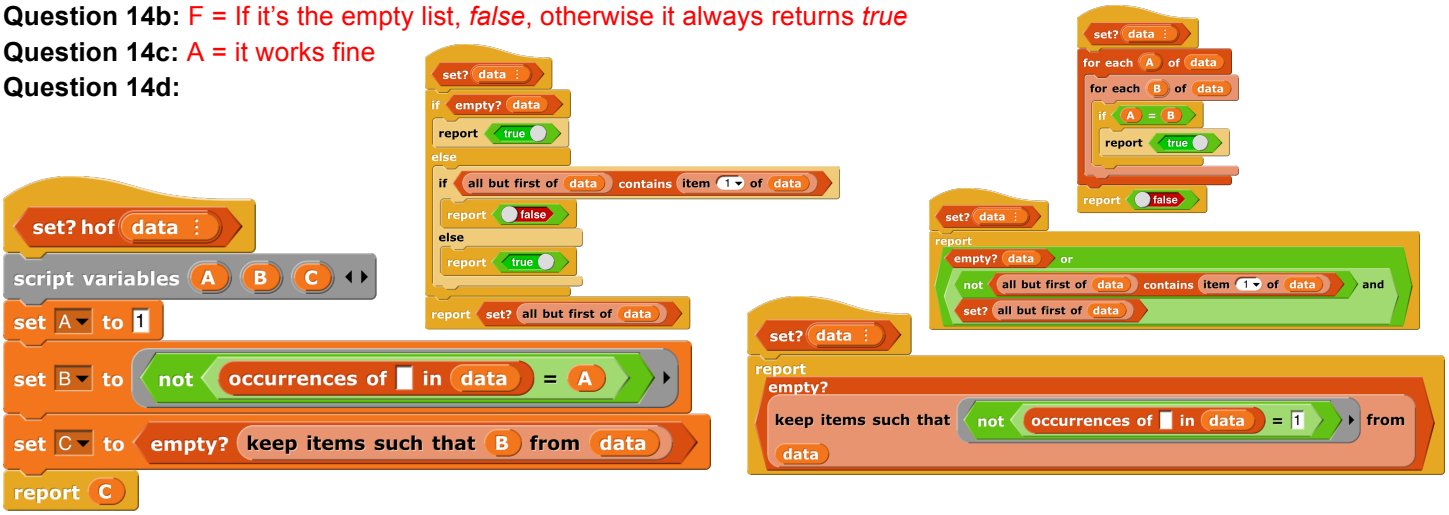
Question 13c: We're told that it actually costs a *dollar* to fill in all the pixels drawn by Helper. Which expression best captures the cost (in dollars) for this call? Based on the L=16, N=3 version, the center Helper is 8 ($8 = L/2) high, and there are 2 4-high (also $8 = L/2), and 4 2-high (also $8 = L/2), which is exactly N copies of that, so it's ½ * L * N.

Question 14a: G = If it's the empty list, *true*, otherwise it only returns whether the *first* element is in the list multiple times

Question 14b: F = If it's the empty list, *false*, otherwise it always returns *true*

Question 14c: A = it works fine

Question 14d:



Question 14e: Quadratic, since it's a linear **keep** of a linear **keep** (how **occurrences** works).

Question 15a: `S = "Berkeley" … S[1:3]` ➔ **er**

Question 15b: `[N ** 2 for N in range(4) if N != 2]` ➔ `[0,1,9]`, because `range(4)` returns `[0,1,2,3]` and the if filters out the **2**, so it's `[0,1,3]`, which get squared by the `**`.

Question 15c: `"".join([word[0] for word in "Univ of Calif at Davis" if not(len(word) == 2)])` ➔ None of these, it's actually "**Univ of Calif at Davis**" because word is actually every character in the string, and word[0] of a single character is itself, so *every* character gets sent to join.

Question 15d: `f1 = lambda x: x+x … f2 = lambda y: y > "9" … list(map(lambda f: f("10"), [f1, f2]))` ➔ `["1010",False]`, because + on strings is concatenation, and strings inside data structures are printed (but aren't when you just print them directly), and > does string comparison (as it does in Snap!, except it doesn't know to convert numbers to number comparison, so "10" appears before "9" in its ordering).

Question 15e: `school` ➔ `"cal" … if school = "berkeley": print("go " + school) else: print("not here")` ➔ This is an error since Python doesn't let you do an assignment in an if expression, but even if you didn't know that, and thought it did work, then we also accepted **go berkeley** (no quotes).

Question 15f:

☑ `define histogram(data):` **def** instead of **define**
☑ `    D = {} ;;; empty histogram ### ` instead of **;;;**
☑ `    foreach item in data:` **for** instead of **foreach**
☑ `        if item in D:` **if item not in D** (if NOT checked, then both `D[item] =` lines must be)
☑ `            D[item] = 0 1` (since when you find one for the first time, you initialize it with 1, not 0)
☑ `        else` **else:**
☐ `            D[item] = D[item] + 1`
☑ `    report D` **return** instead of **report**