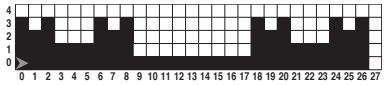
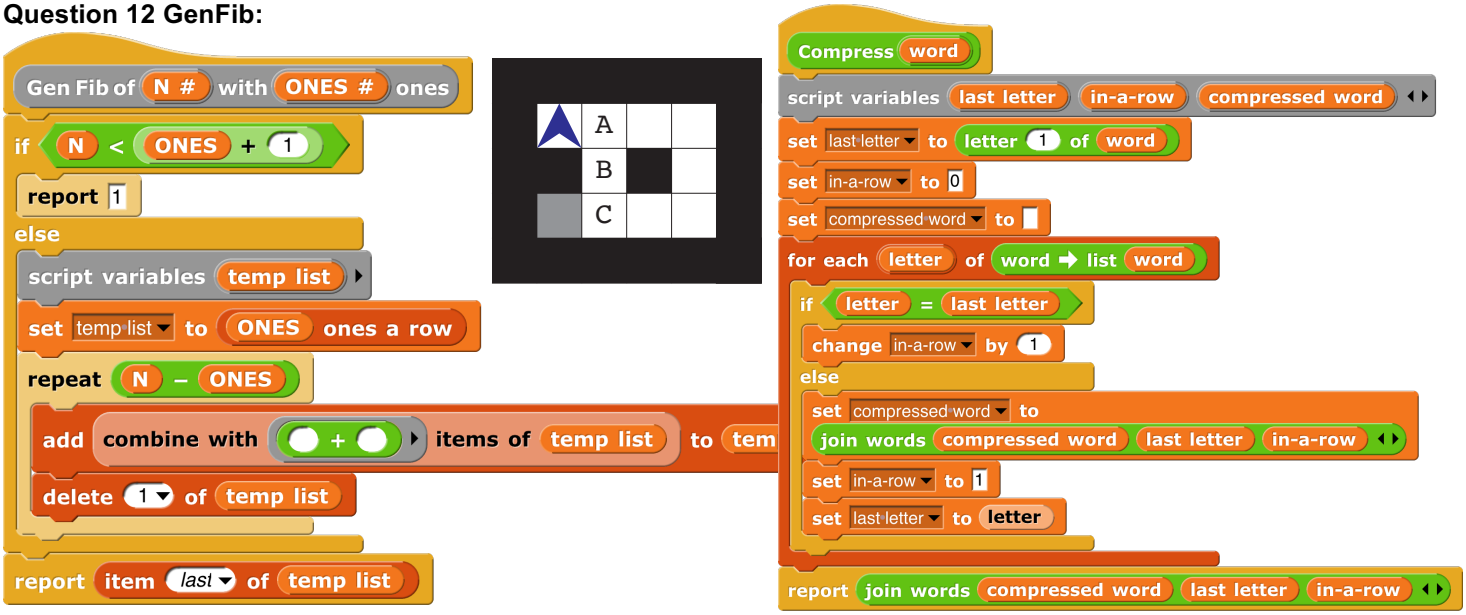
CS10 Spring 2017 Final Exam Answers

Question 1: Free points for attending the alumni panel! What happened? Michael Ball came late (from the side door)
Question 2: Which of the following is *true?* None of the above. Moore's law talks about number of transistors on an integrated circuit, and it's still chugging along. Most of the time in scientific code is spent moving data around!
Question 3: Around 1990, what happened in the world of computing (from an HCI viewpoint)? Ubiquitous computing era.
Question 4: If today someone proved that P *did not* equal NP, that'd mean for the *FIRST TIME*… *None of the above. It would prove that there are "easy" (P) problems and "hard" (NP) problems, and we can finally stop trying to search for hidden P algorithms to the latter, because it will have been proved that they don't exist – they are fundamentally too hard.*
Question 5: Internet standards started moving to IPv6 (and away from IPv4) because… None of the above. It's because we needed more address space, which none of these options address.
Question 6: How much better does the *optimal* solution do over the *greedy* solution? Greedy: $15@15lbs (then no more space); Optimal: Two $6@7lbs, One $5@6lbs for a total of $17@20lbs, so $2 more.



Question 7: Mystery draws the Cantor Fractal. The LEVELS=1 is just the bottommost two rows. This is LEVELS=3.
Question 8: How many values of NUMBER are there at the end (given **+**1, re**V**erse, **R**ight)? It's basically the permutation of the three blocks: +→V→R=142, +→R→V=214, V→+→R=232, V→R→+=133, R→+→V=313, and R→V→+=214(5 total)
Question 9a: `mycombine(joinswap,[1,2,3])` if **randint** always returns 0 (i.e., always picks the *leftmost* index)? 321.
Question 9b: `mycombine(joinswap,[1,2,3])` if **randint** always returns `len(L)-2` (i.e., the *rightmost* index)? 321.
Question 9c: `mycombine(minus,[4,3,2,1])` → (4-3)-(2-1)=**0**, 4-((3-2)-1)=**4**, (4-(3-2))-1=**2**, 4-(3-(2-1))=**2**, ((4-3)-2)-1=**-2**
Question 10a: How many *total mazes* can we create? 3 bits is 8 mazes.
Question 10b: How mazes are *solvable*? A and C have to be empty, B can be anything, so 2.
Question 10c: What expression is *true* for all the solvable mazes? A and C have to be true (B anything), so `A and C`.
Question 11: ***Algorithm I "go down the line":*** Linear running time and comparisons. ***Algorithm II "tournament":*** Logarithmic running time (since the comparisons are done simultaneously in parallel) and linear comparisons. ***Algorithm III "everyone check everyone else":*** First one checks n-1, next checks n-2, next checks n-3, …. The sum of 1 to n is half of a square, so it's $n^2/2$, or Quadratic in running time and comparisons. ***Algorithm IV: "everyone check everyone else":*** If the first person is not the biggest and is chosen over and over again, it could go forever with ∞ comparisons.
Question 12 GenFib:



Question 13a: Move "set last letter to letter" outside if. Nothing changes: compress(a)→a 1, compress(abba)→a 1 b 2 a 1.
Question 13b: Change 

to 

.compress(a)→a 0 a 1, compress(abba)→a 0 a 3 a 1 (just have to trace it slowly).
Question 13c: Your friend suggests to compress the return even further, and instead of using **join words** to build up the compressed word, use **join** (so there will be no spaces in the output). Thoughts? There's a problem, since you can't uniquely uncompress all numbers (a special kind of "word" with digits) For example compress(111…1) [234 times] →1234, and we don't know whether that was 234 "ones" OR 2 "one" and 4 "threes".