# Programming Test

1. (1 point) Which of the following data structures is typically laid out in memory with elements addressed at sequential addresses.
2. A binary tree
3. A pointer
4. An array
5. A linked-list
6. None of the above
7. All of the above
8. (1 point) Which of the following data structures can have elements added to or removed at either end, but not in the middle:
9. An array
10. A linked-list
11. A stack
12. A queue
13. None of the above
14. All of the above
15. (1 point) How do I compute the address of an element in an array: (no idea)
16. element\_address = base address of the array \* element\_index
17. element\_address = base address of the array + element\_index
18. element\_address = base address of the array + (size\_of\_an\_element \* 3) + 1
19. element\_address = base address of the array + (element\_index \* size\_of\_an\_element)
20. None of the above
21. (1 point) When searching for a desired value in a sorted list, how does a “Divide and Conquer” algorithm work?
22. select each element in the list, starting at the first element, and decide if the element is larger or smaller than desired value
23. starting at the end of the list, check elements in pairs for a match on the desired value
24. starting in the middle of the list, decide if the element value is larger or smaller than the desired value, and then recurse, using the half of the list until a match is found
25. divide the elements in the list by 2, and see how many are still greater than 1, and recurse until one element remains
26. None of the above
27. (1 point) When frequently adding and deleting elements in a sorted data structure, what is the most efficient data structure to use?
28. A matrix
29. An array
30. A linked list
31. A tree
32. A string
33. (1 point) When searching for data in an n-element sorted array, what is the maximum number of operations you will need to perform when using a Binary Search algorithm? (logn)
34. n2
35. nlog2n
36. n3
37. n/2
38. None of the above
39. (1 point) What is the key property of a max heap?
40. The middle element is the smallest element in the heap.
41. The first element is largest element in the heap.
42. The last element is the small element in the heap.
43. The elements are sorted in descending order.
44. None of the above
45. (2 + 2 = 4 points) Write programs in a high-level language or pseudocode for the following two problems. You will not be graded on the syntax of your program, as long as the program description is clear.
    1. Given a list L with n integers, return a new list M which is the reverse of L.

reverse = list()

for n in range(0,len(L)):

reverse.append(L.pop())

* 1. Given a list L with n distinct integers, return the element of L that has the second largest value.

sorted(L)[-2]

Do you think using python functions like sorted and pop is cheating?

1. (2 points) In words, describe what function does the program QuizFunction() implement?

Assume that A and B are lists of integers. The function len(X) returns the length of list X and list(k) returns a list of length k in which each element is set to zero.

defQuizFunction(A, B):

if len(A) <len(B):

C = list(len(A))

for i from 0 to len(A):

if A(i) < B(i):

C(i) = B(i)

else:

C(i) = A(i)

else:

C = list(len(B))

for i from 0 to len(B):

if A(i) < B(i):

C(i) = B(i)

else:

C(i) = A(i)

Creates new list with length of longer of two options. The element at index x of new list is the greater of two options at index x. Longest list and biggest elements

-

1. (3 points) Consider the following pseudocode snippet. What will the following function calls return?
   1. recursive(7,3) 12
   2. recursive(7,4) 1
   3. recursive(7,-1) -1

def recursive(x, a):

if a is 0:

return 1

if a is odd:

b = recursive(x,(a-1)/2)

return a\*b\*b

else:

b = recursive(x,a/2)

return b\*b

1. (1 + 1 = 2 points) Describe in words what the following common Linux would output?
2. ls – list directory contents
3. grep “and” names.tex – looks for word “and” in the names.tx file?