

# Programming And Data Structures – Fall 2019

**This test contains 12 questions for a total of 17 points + 1 bonus point.**

You are **not allowed** to use any form of calculators in this exam!

1. (1 point) You want to form a team of 5 players out of 10 candidates. How many distinct teams can you make?

- A. 252
- B. 30240
- C. 525
- D. 120

2. (1 point) Which of the following is **greater** than the others? (you don't need a calculator! Just apply exponentiation and logarithm rules)

- A.  $2^6 * 4^8$
- B.  $(\log_2(4^{16})) * 2^{17}$
- C.  $2^{\log_2(4^{32})} * 2^{16}$
- D.  $(\log_2(1024 * 512 * 256 * 32)) * 2^{15}$

3. (1 point) Which of the following operations **cannot** be done in constant time for an array of **sorted data**. Assume that array elements are distinct.

- A. Find the i'th largest element
- B. Delete the i'th largest element
- C. Find the i'th smallest element
- D. All of the above

4. (1 point) Assume you have a stack of 100 elements. How many pop operations should you perform **before** you are able to pop the element at the bottom of the stack?

- A. 100
- B. 99
- C. 1

D. 0 (there is no need to do pop operation)

5. (1 point) Which data structure would you use to store the mapping between customers' IDs and their full names as key and value pairs, which have on average constant time for lookup and insertions?

- A. Stack
- B. Key Array
- C. Hash Table
- D. Tree

6. (1 point) What is the return value of the following function, when the input argument is `nums=[1,2,3,4,5]`? (assume `length` function returns the length of the given array)

```
compute(nums) {  
    x = 0  
    n = length(nums)  
    for(i = 0; i < n; i++) {  
        x = x + nums[i]  
    }  
    return x/n  
}
```

- A. 15.0
- B. 4.0
- C. 3.0
- D. 5.0

7. (2 points) What is the run time complexity of the following code?

```
for(i = 0; i < n; i++) {  
    j = 0  
    while(j < n) {  
        print(i+j)  
        j = j + 1  
    }  
}
```

- A.  $O(n^2)$
- B.  $O(2n)$
- C.  $O(i+j)$
- D.  $O(i*j)$

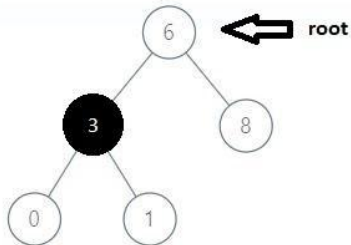
E.  $O(n)$

8. (2 points) Assume you have a binary tree structure as defined by the following `TreeNode` class. What will be the value of the left child of **the black node** in the following picture after calling `flip` function on the **root** node of `myTree`?

```
class TreeNode {
    value //The integer value that this node holds
    left  //The node which is the left child of this node
    right //The node which is the right child of this node
}

flip(t) { //t is an instance of a TreeNode
    if(t.value % 2 == 1) {
        temp = t.left
        t.left = t.right
        t.right = temp
    }
}
```

myTree:



- A. 0
- B. 1
- C. 8
- D. 3
- E. 6

9. (2 points) Consider the following definition for `calc` function. What is the return value of this function when `x = 4`?

```
calc(x) {
    if(x < 3) {
        return x
    }
    return calc(x-1)*calc(x-2)
}
```

- A. 3
- B. 4
- C. 12
- D. 8

10. (2 points) What will be the output of the following function, when the input arguments are two arrays as `nums1=[1,2,10]` and `nums2=[3,9]`. (Assume `list[m+n]` creates an array of length `m+n`)

```
someFunction(nums1, nums2) {
    m = length(nums1)
    n = length(nums2)

    result = list[m+n]

    i = j = k = 0

    while(i+j < m+n) {
        if(i < m and j < n) {
            if(nums1[i] < nums2[j]) {
                result[k] = nums1[i]
                i = i + 1
            } else {
                result[k] = nums2[j]
                j = j + 1
            }
        }
        else if(i < m) {
            result[k] = nums1[i];
            i = i + 1;
        }
        else {
            result[k] = nums2[j];
            j = j + 1;
        }
        k = k + 1;
    }
    return result[m+n-1]
}
```

- A. 1
- B. 2
- C. 9
- D. 10
- E. 3

11. (1 point) This is a bonus question, you may leave the answer blank. What will be printed to console when you run a program on your computer which has the following line?

```
print((0.1+0.2) == 0.3)
```

12. (3 points) Write a program in a high-level language or pseudo-code for this problem. You will not be graded on the syntax of your program, as long as the program description is clear.

Write a function which finds and returns the **element with the highest frequency** in an integer array  $M$  of size  $n$ . You can assume that there will be a unique element in  $M$  which has the highest frequency and we have  $0 \leq M[i] \leq 100$  for all the elements in  $M$ .