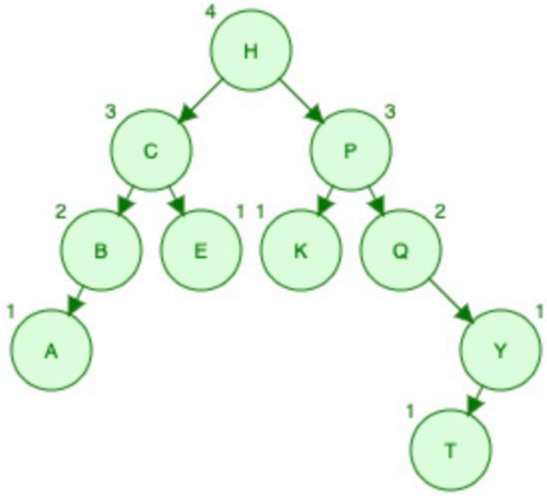
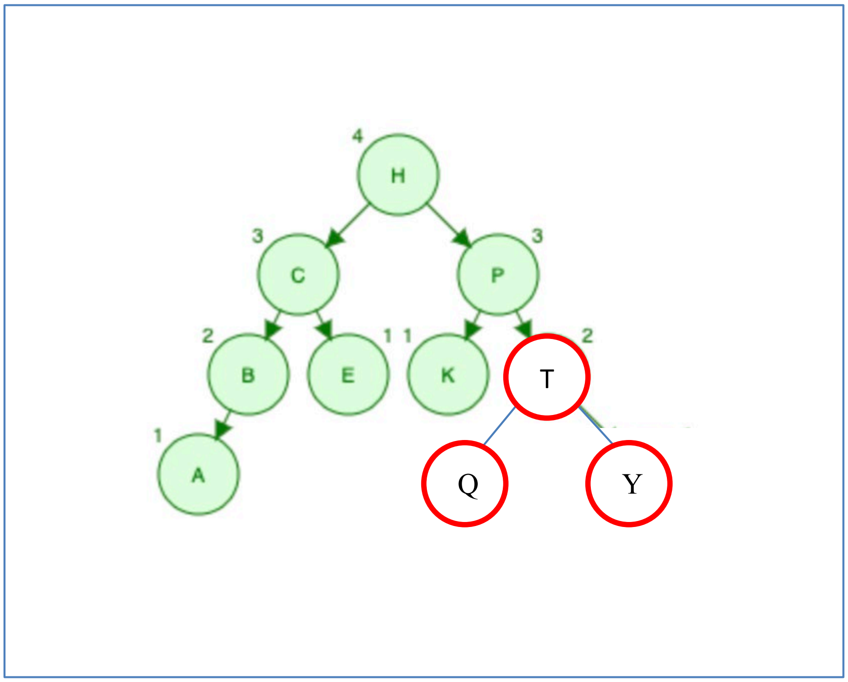
TIC2001 Data Structure and Algorithm Quiz 2
Name: ______________ Student#:_A_____________

Question 1 (5 × 3 + 5 marks)

This is an AVL tree after inserting one alphabet into a height-balanced tree but before balancing. For alphabet, we assume 'A' < 'B' < 'C' < .. < 'Z.





The newly inserted node is the letter _T____
And the lowest node that is not height balanced is the node with letter ___Q___
And the number of rotation(s) to restore the height-balance is __2____
Please draw the tree after balancing at the right of the given tree (5 marks)

Question 2 True or False (9 x 3 marks) (Put your answer at the right column)

	Answer:
The worst time complexity of in-order traversal for a binary tree is O(n log n)	F
For a binary search tree, if y is the successor of x in the tree, x can never have two children.	F
Division Method is faster than Multiplication Method in general in hashing because Division Method does not need to compute large numbers.	F
In hashing, Linear Probing is preferred over Quadratic Probing in practice because Linear Probing is faster in calculation	F
For AVL trees, the number of rotations after deletion is O(log n).	T
Assuming the size of hash table is m with n items inserted. If m is a lot larger than n, the time complexity of search for a successor in the hash table is O(m).	T
With open addressing and linear probing, first, we insert n items into a hash table. Then we delete n-1 items from it. To search for the remaining item in the hash table, it is possible to take O(n) time	T
The height of a binary tree can be O(n) if there are n items in the tree.	T
The best way for order statistics is to store the rank of the node in each node	F

Question 3 (10 + 3 marks)

Given a hash table with size m = 13 buckets. And the hash function h(string) = square of the letter “number” of the **second** character with letter number of a = 1, b = 2, etc.. For example:

$$h(\text{“h}\underline{\text{a}}\text{t”}) = 1 \times 1 = 1. \text{ (Because “a” = 1)}$$

With i equals to the number of collisions, perform hashing with **Quadratic** probing for the key x with the probing,

$$(h(x) + i^2) \bmod m.$$

with **open addressing**.

Question 3a

Update the hash table on the right according to the commands on the left similar to the exercise we did in the lab sessions.

Commands:

```
insert(“durian”)
insert(“apple”)
insert(“orange”)
insert(“mango”)
insert(“milk”)
insert(“salt”)
insert(“cake”);
delete(“salt”)
insert(“icecream”)
delete(“cake”)
insert(“pineapple”)
delete(“mango”)
insert(“cake”)
```

Index	Hash Table Contents					
0	Null	orange(1)		5		
1	Null	~~mango~~	cake	0		
2	Null	~~salt(1)~~		1		
3	Null	milk				
4	Null	pineapple(1)		4		
5	Null	~~cake(2)~~		2		
6	Null					
7	Null					
8	Null					
9	Null	apple				
10	Null	icecream(1)		3		
11	Null			(banana)		
12	Null	durian				

Note that it is not chaining. The space on the “right” of “Null” is just for your “replacement” of the contents. You just need to cross out a word if it’s deleted.

Question 3b

After all the above operations, if we ***search*** for “banana”, what is the index of the ***last*** hash table entry that the searching visited before telling us that “item not found”?

The index of the last hash table entry = _______11_________________.