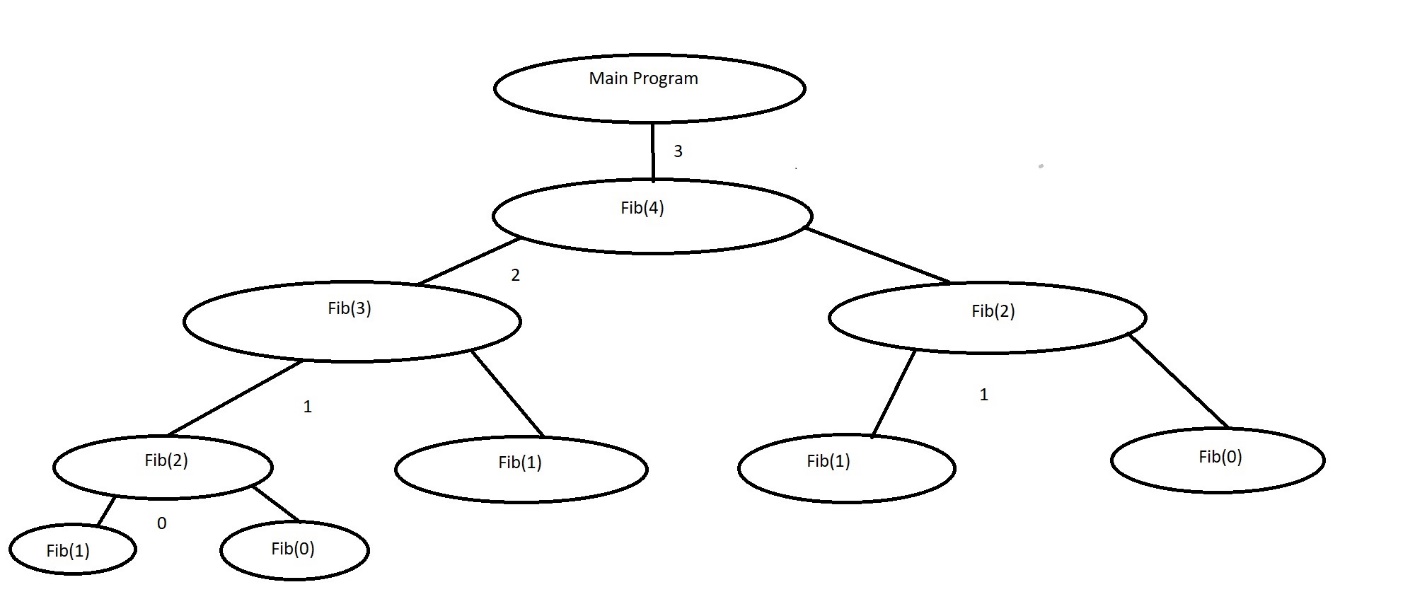
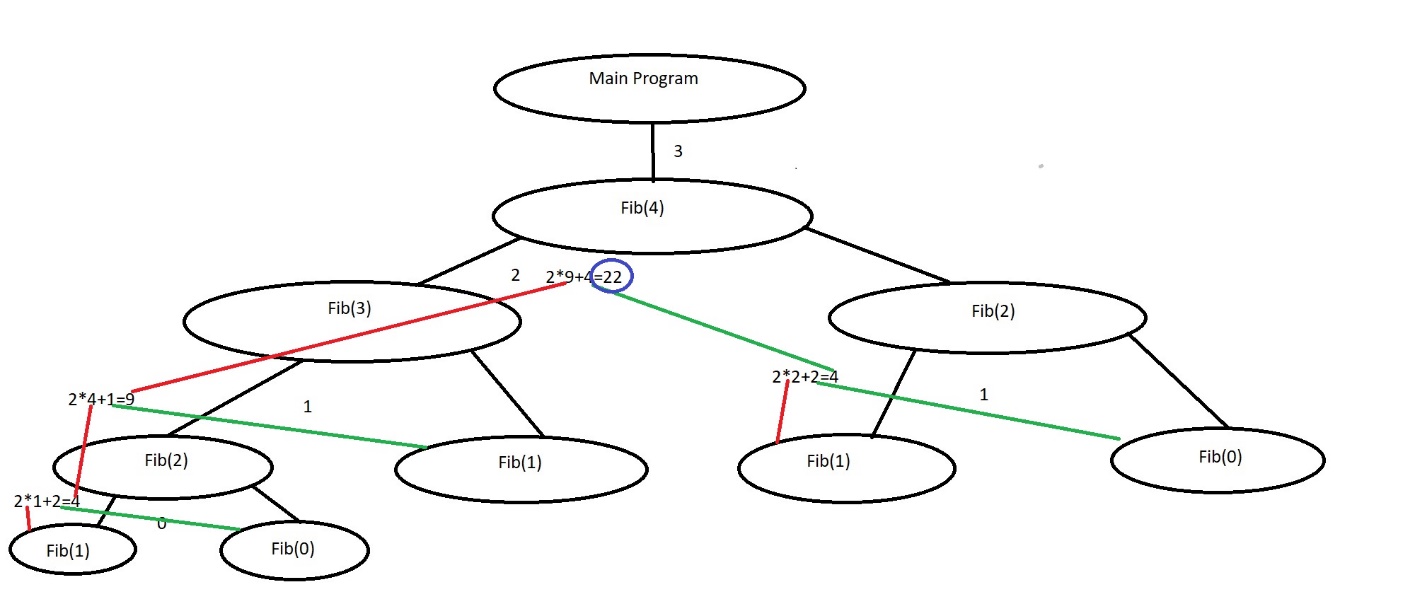
1.

a.

**Fib is called 9 times.**

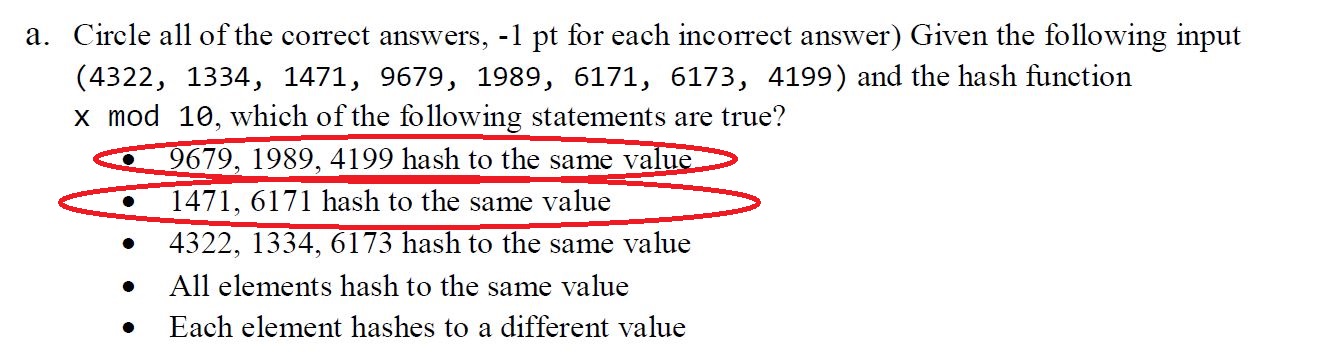
b. 

**The Number returned is 22.**

c. The output is located in transcript.txt

and my code is available in maxOfList.c

2.



When you mod each number, you’re dividing except the answer is only the remainder. SO when you mod the numbers by 10, all of the number is ignored except for the number in the one’s place (the right most part of the number). Therefore, the numbers with 9 in the one’s place all group together, the numbers with 1 in the one’s place all group together and the other numbers hash alone to their own address.

b.

When using a hash map, sometimes the items are sent to the same location in memory, when that happens the items need to be dealt with so that they can be recovered. We use techniques called conflict by resolution which are chaining and open addressing.

Conflict resolution by Chaining is when more than one items sent to the same address of the hash map share the spot and are linked together by either a linked list or an array. The problem with this is that the lists or arrays can get very big if you’re dealing with a lot of data that has a lot of collisions. If the lists are long they have to be searched sequentially until the data is found which can make the Big O of the hash map worse.

Conflict resolution by open addressing is when the data, after having a collision with the hash map, sends the data to the next available spot in memory by use of some collision function. The problem with open addressing is that it can fill a spot that another piece of incoming data could get hashed too, as well as the complexity of finding a new spot if the data keeps moving to spots that are full.

3. done in stacksAndQueues.exe, and the transcripts can be viewed in SaQtranscripts. I completed the extra credit.