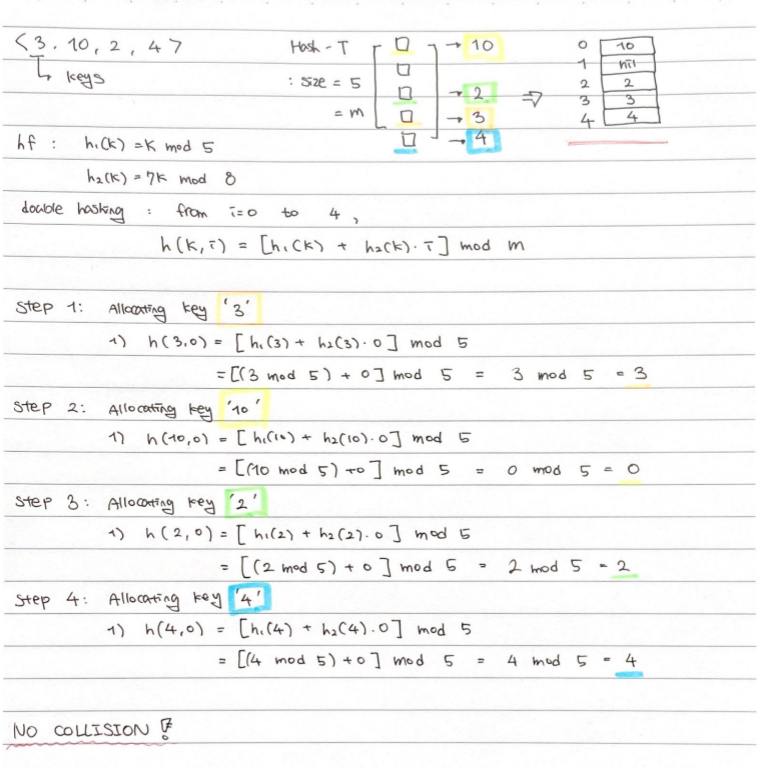
Assignment 10

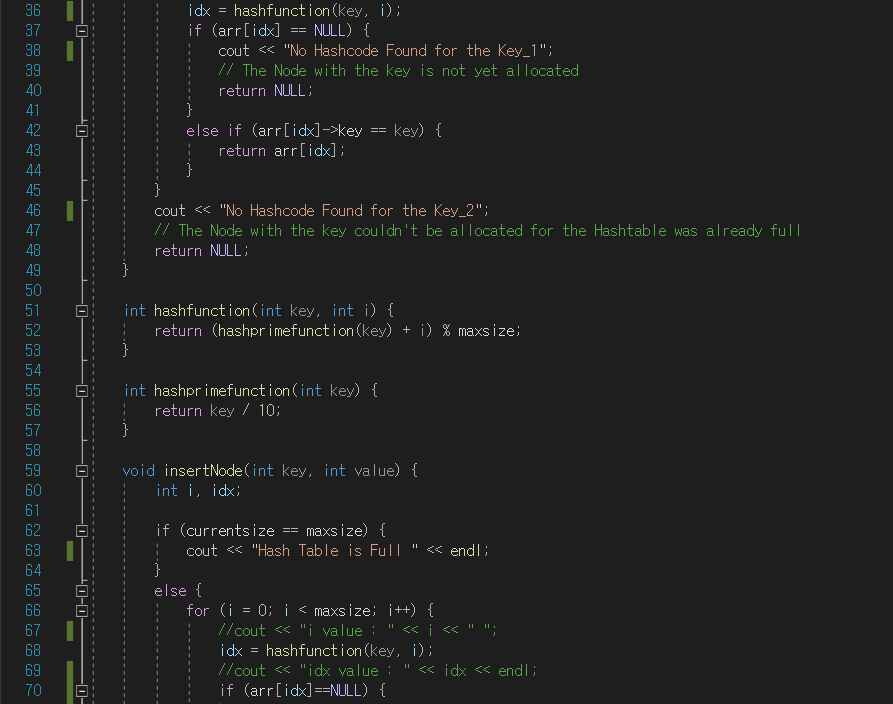
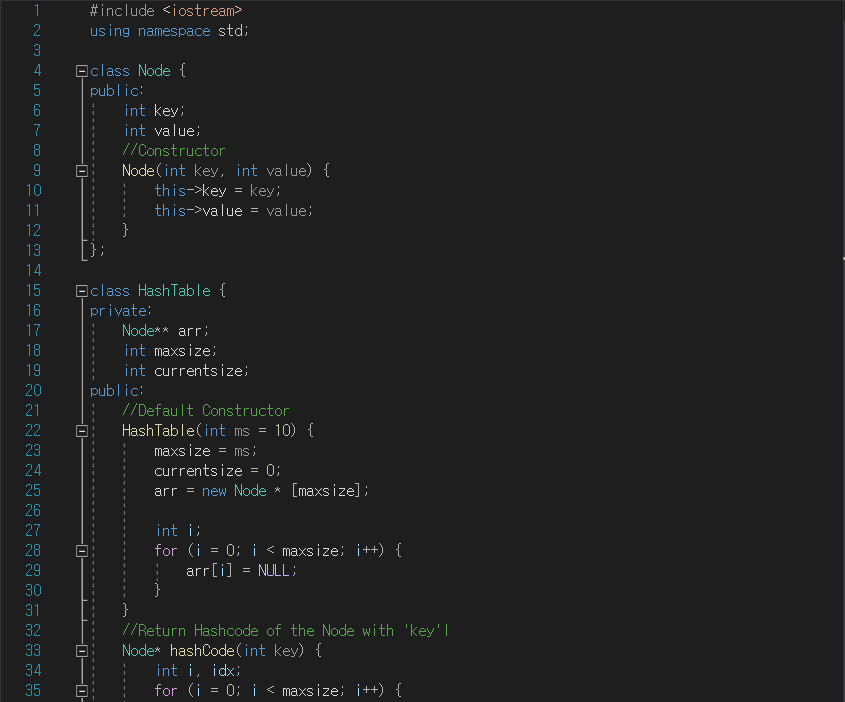
Name: DONGWOOK LEE

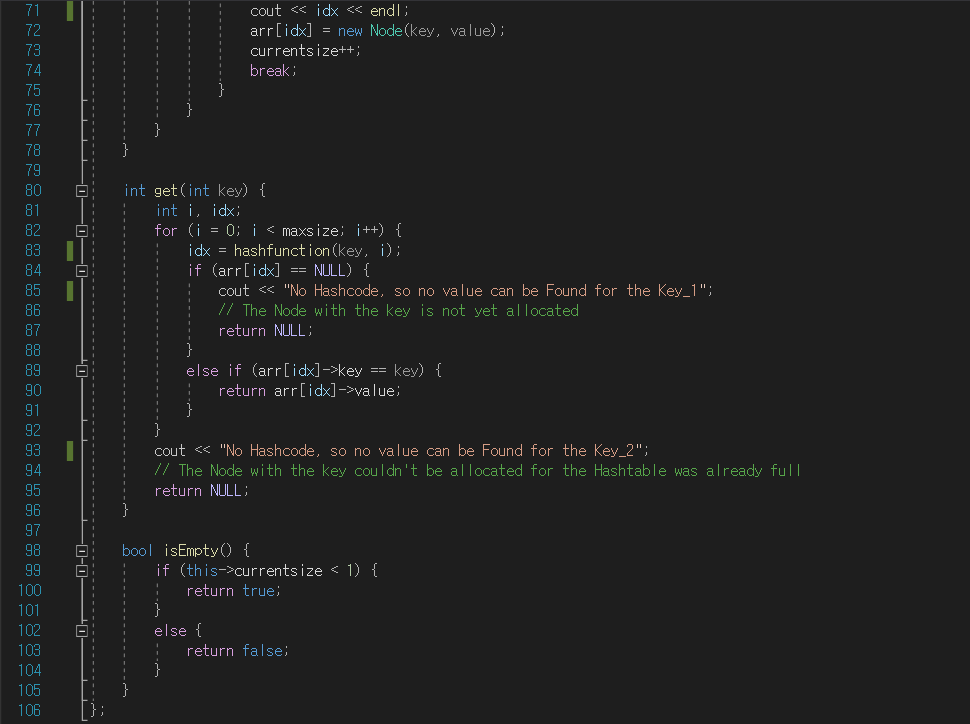
**Problem 1:** *Hash Tables*

a) (4 points)

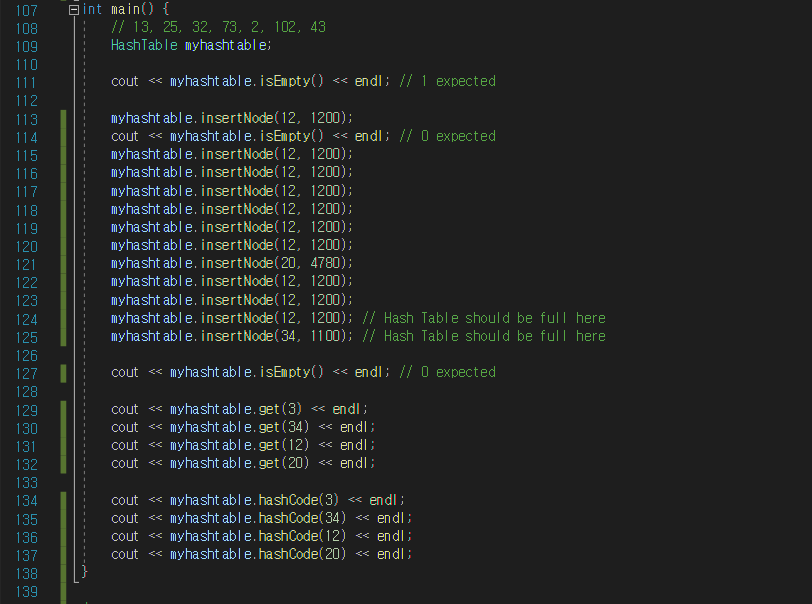
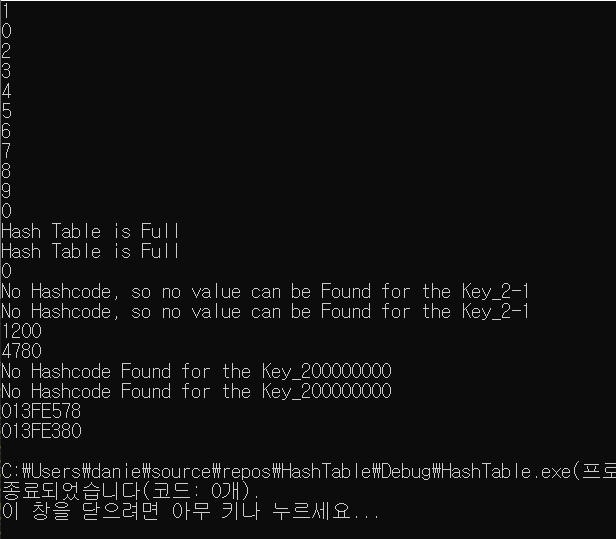


b) (7 points)





EX 1)

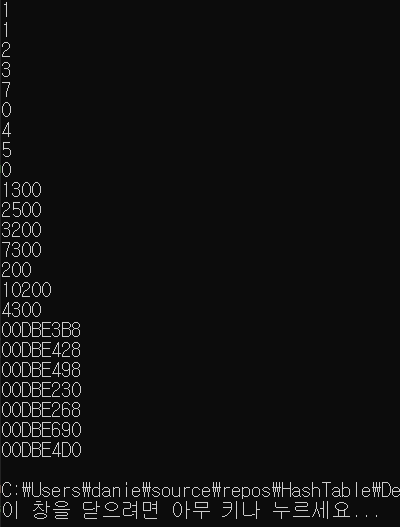
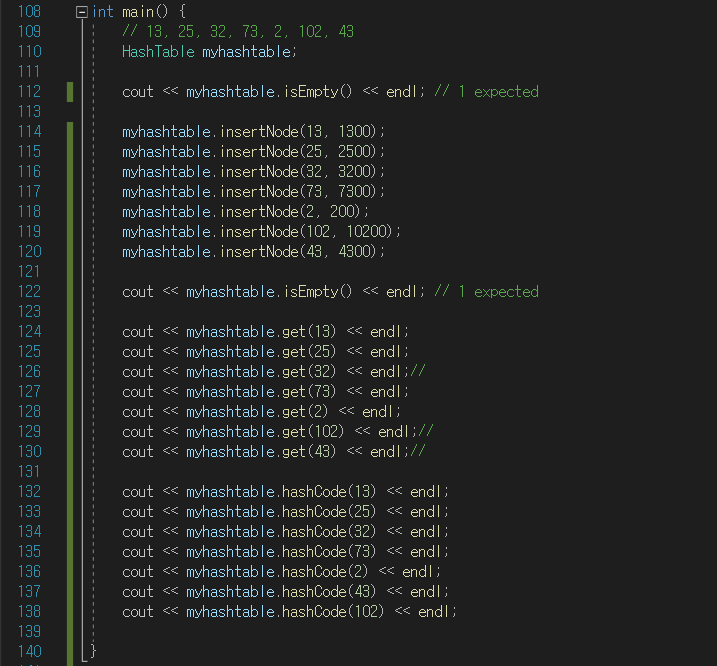
 

\* It prints out 00000000 and -1 at the end of each error sentence because

Node hashcode() function returns NULL as 8 bits

and int get() returns -1 in an error case.

EX 2)



Assume I have a test data set A(key) = {13, 25, 32, 73, 2, 102, 43}

If I create a default Hash Table using constructor, my code will create a Hash Table with size of 10.

It means, when I use Linear Probing hash function, the number in the 1’s digit of each element

in A will consider where to place the element in my Hash Table.

Accordingly, if there exist more than one element with same number in 1’s digit, there will be a

collision.

However, if I `div` each of element in set A by 10, they become {1, 2, 3, 7, 0, 10, 4}.

Now the elements will not have any collision even if we calculate `mod` of each of them.

Therefore, my h’ function to be h’ = key / 10 will yield efficient allocation of the elements into my

Hash Table.

**Problem 10.2** *Greedy Algorithms*

a) (2 points)

