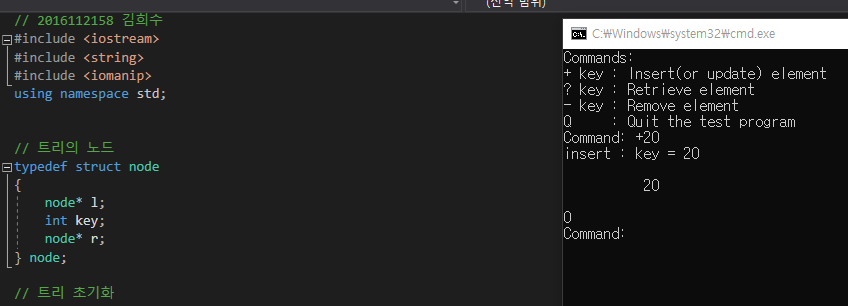


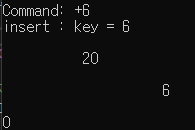
6주차 알고리즘과제

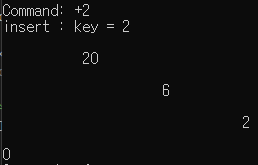
2016112158 김희수

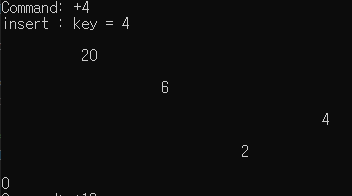


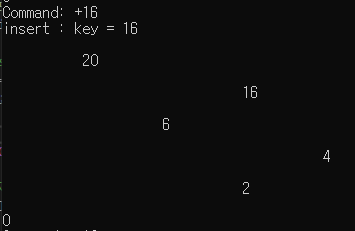
(20,6,2,4,16,10,8,12,14,9 삽입과정)

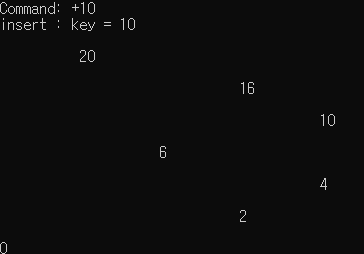








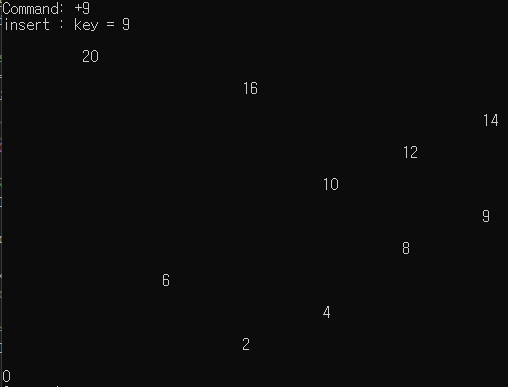




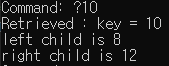




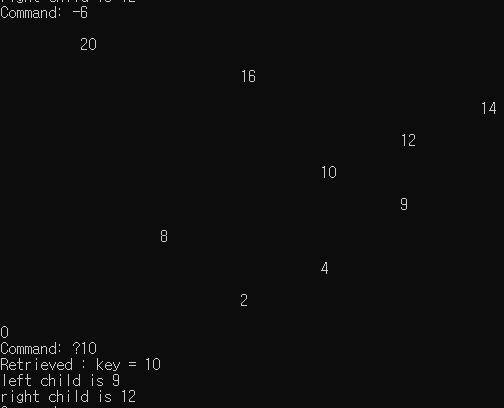




(T1에서 10을 탐색하고 10의 오른족, 왼쪽 노드의 키값 출력)



(T1에서 6을 삭제하고, 10을 탐색하여 10의 오른쪽, 왼쪽 노드의 키값 출력)



treeInit()함수는 처음에 트리를 생성하는 함수이다.

treeSearch()함수는 트리의 head(root)와 찾고하자하는 키를 매개변수로 받아 탐색을 수행하는 함수이다. 키를 찾으면 리턴하고, 찾고자하는 키가 현재키보다 작으면 왼쪽에서 탐색, 크면 오른쪽에서 탐색을 계속한다

treeInsert()함수는 키를 삽입될 트리와 삽입될 키를 매개변수로 가져와 트리에 키를 삽입하는 함수이다. 새로운 노드를 생성하여 삽입한다. root의 키와 삽입될 키의 대소비교를 통해 재귀적으로 수행된다\_

treeDel()함수는 트리와 트리에서 삭제될 키를 매개변수로 받는다. 먼저 recursion의 끝이될 base case를 설정한다. 지우려는 키와 root의 키의 대소비교를 통해 재귀적으로 함수가 실행된다. 자식노드가 없거나 하나만 있다는건 삭제하려는 노드가 단말노드이거나 한 개의 서브트리를 가진다는 거다. 단말 노드의 부모노드를 찾아 연결을 끊거나, 연결을 끊고 서브트리는 삭제된 노드의 부모노드에 붙여주면 된다. 삭제하려는 노드에게 자식노드가 모두 있는 경우, 오른쪽 서브트리의 최소값을 갖는 노드를 찾아서 이를 루트에 연결해주면 된다.

print2DUtil()함수는 트리를 출력해주는 함수이다. 오른쪽 자식 서브트리와 왼쪽 자식 서브트리에 대해 각각 연산을 수행해주면 되므로 재귀적으로 구현되었다.

그 후 main 함수에서 treeInit()를 통해 트리를 생성한 후 whlie문을 통해 Q를 입력하지 않는한 중지되지 않도록 만든다. 이를 위해 while중지를 위한 flag가 사용된다. 그 후, input를 cin으로 입력받고 command의 첫글자는 트리의 연산을 위한 \_operator로 설정한다. switch-case문을 이용해 \_operator에 따라서 다른 연산이 수행된다. 변수 key는 input의 첫글자를 제외한 부분, 즉 연산자를 제외한 “값”이다.