

다음에 열거된 요구사항(requirement)를 충분히 숙지하고 과제를 제출하시오.

1. Binary Search Tree의 기능 구현 #2에 관한 것이다.

(a) binary-search-tree-2.c의 다음 함수를 완성한다.

```
void recursiveInorder(Node* ptr);      /* recursive inorder traversal */
void iterativeInorder(Node* ptr);      /* iterative inorder traversal    p.224 */
void levelOrder(Node* ptr);           /* level order traversal          p.225 */
int deleteNode(Node* head, int key);   /* delete the node for the key: 3 cases */

Node* pop();                          /* for stack */
void push(Node* aNode);

Node* deQueue();                      /* for queue */
void enQueue(Node* aNode);
```

(b) deleteNode(Node* head, int key) 함수는 총 3가지 사항에 대해 구현되어야 한다.

- 삭제하고자 하는 노드가 단말노드 일때
- 삭제하고자 하는 노드가 하나의 자식만을 가질 때
- 삭제하고자 하는 노드가 두개의 자식을 가질 때. 이 경우 **오른쪽 서브트리에서 가장 작은 값으로 대체되도록 한다.**

(c) 이해한 부분을 주석으로 남긴다.

2. GNU C Compiler + Open Source Editor를 사용한다.

3. 소스파일에 성명, 학번이 실행시 출력되도록 한다(sprintf()).

[----- [Your Name] [Student ID] -----]

4. **GitHub에 hw10 Repository를 생성하고 binary-search-tree-2.c를 업로드 한다.**

5. **소스코드를 최소 3번 Commit하여 진행과정을 GitHub에 기록한다.**

6. **작성된 소스파일을 보고서로 만든다.**

7. 보고서에 실행결과를 Screen Capture하여 첨부한다.

8. 과제 결과물을 eCampus에 업로드한다.

9. **주의사항**

- 마감시간을 넘긴 모든 과제는 0점 처리됨.**
- 주석이 부실한 코드는 감점 대상이 되며, 프로그램 복사의 경우 모두 0점 처리됨.**
- 보고서에 소스 코드를 반드시 첨부할 것.**

- (d) GitHub를 이용하지 않은 과제는 0점 처리됨.
- (e) 코드가 아닌 **과제 단위**로 GitHub Repository를 만들것.
- (f) 컴파일(GNU C Compiler) 및 실행이 안될 경우 0점 처리됨.
- (g) 정상동작하지 않은 프로그램 감점 처리됨.