처음 코드를 실행하면 pointer, array 중 선택을 할 수 있게 하였다.

#POINTER

pointer을 선택할 경우 pointer에서 구현 가능한 함수들을 a~h번 중에서 선택할 수 있으며, q번을 선택할 경우 종료할 수 있게 만들었다.

Data, left pointer, right pointer를 변수로 갖는 구조체 노드를 만들어 코드를 짰다.

>insert

값을 입력 받아 BST를 형성하도록 만들었고, 같은 값을 입력 받으면 ‘이미 같은 값이 있다’는 문구를 반환한다.

>delete

자식 노드의 개수에 따라 삭제하는 코드를 짰고, 노드에 데이터 값이 없을 때 데이터 값이 없다는 문구를 반환한다.

이때 자식 노드가 두개일 때 오른쪽 자식 노드 구역에서의 최솟값을 가져오기 위해 findmin함수를 이용했다.

>inorder preorder

좀 더 트리를 명확히 확인하기 위해 preorder도 구현해봤다.

>find children

자식의 유무, 자식의 수, 자식의 위치 모두 찾을 수 있도록 구현했다.

>find parent

Parent를 찾고자하는 데이터를 찾아나가면서 찾았을 때 그 데이터를 가리키는 부모의 포인터값을 반환하는 함수이다.

#ARRAY

array를 선택할 경우 array에서 구현 가능한 함수들을 a~g번 중에서 선택할 수 있으며, q번을 선택할 경우 종료할 수 있게 만들었다.

이때 배열의 크기는 100으로 설정하였고, 처음에 배열을 초기화 시켜주기 위해서 유저가 입력하지 않을만한 작은 수를 N=-100000으로 잡고 코드를 짰다.

주된 아이디어는 원래 노드가 배열의 k번째에 위치한다면 left child는 2k+1, right child는 2k+2에 위치한다는 개념을 사용했다.

>insert

값을 입력 받아 BST를 형성하도록 만들었고, 같은 값을 입력 받으면 ‘이미 같은 값이 있다’는 문구를 반환한다.

>delete

구현하지 못했다.

>inorder preorder

좀 더 트리를 명확히 확인하기 위해 preorder도 구현해봤다.

>find children

자식의 유무, 자식의 수, 자식의 위치 모두 찾을 수 있도록 구현했다.

>find parent

Parent를 찾고자하는 데이터를 찾아나가면서 찾았을 때 그 데이터를 가리키는 부모의 포인터값을 반환하는 함수이다.

다음은 컴파일 후 콘솔화면에서 실제로 구현해본 결과이다.

