**README file**

1. void buildHeap

makeHeap 함수를 이용하여 구현하였다. makeHeap 함수는 input이 되는 array가 max-heap property를 만족시키도록 하는 함수이다. makeHeap 함수에 array를 넣고 heapData에 array를 할당해서 구현했다. Heap이 NULL인 경우, 에러를 출력하도록 했다.

2. void deleteHeap

dequeueHeap함수를 h의 size 값만큼 반복해서 heap을 delete하도록 구현하였다. dequeueHeap에 free(h)를 작성하여 h를 free시켰다. Heap이 NULL이거나 depth가 0일 때 에러를 출력하도록 했다.

3. int findDepth

Heap의 size에 log2와 floor함수를 취하고, root node의 depth가 1이어야 하므로 1을 더해서 heap의 depth를 계산했다. 계산한 depth 값을 return하여 depth를 출력하도록 하였고, heap이 NULL이거나 size가 0일 때는 에러를 출력하도록 했다.

4. void insertNode

H의 size를 1 늘리고 heapData의 마지막 노드에 value를 추가한다. 그리고 makeHeap을 실행하여 새로운 노드가 들어온 h가 max-heap property를 만족하도록 구현했다. H가 NULL이거나 size가 0일 때는 에러를 출력하도록 했다.

5. int dequeueHeap

Heap의 root의 값과 마지막 node의 값을 바꾼다. 그리고 마지막 node의 값을 0으로 할당하고 heap의 size를 하나 줄인 후 makeHeap 함수를 실행해 바뀐 h가 max-heap property를 만족하도록 하고 기존 root 값은 return하여 출력하도록 구현했다. Size가 1일 때는 dequeue를 실행하고 나면 heap이 존재하지 않으므로 free(h)를 넣어서 h를 free시켰다. Heap이 NULL이거나 depth가 0이면 에러를 출력하도록 했다.

6. void heapsort

Min-heap을 만드는 make\_min\_Heap 함수를 이용해서 구현했다. 주어진 array를 min heap으로 만들고 dequeueHeap을 하면 array에서 오름차순으로 return값을 얻게 되는데 이를 heapData에 순서대로 할당하였다.