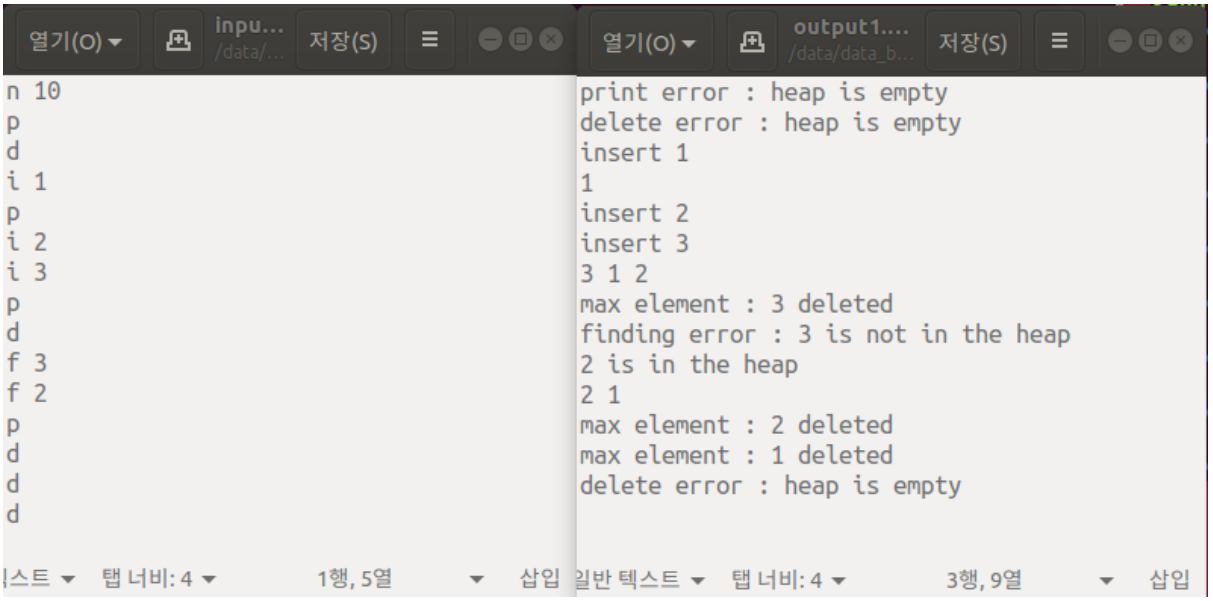
2023 Data Structure (CSE2010 12282)

Lab 08 (5/04) Max Heap

- arranged by TA Sehyun Cha

Due : ~2023.05.10.(Wed) 23:59, Late Submission : ~2023.05.11.(Thu) 23:59

- **Max Heap Implementation - Create, Insert, DeleteMax, find, PrintHeap**
- 아래의 사진과 같이 **[input].txt** 파일을 입력받아 **Max Heap**을 생성하고 편집하는 기능을 구현하고 결과값을 **[output].txt** 파일에 출력하여 저장한다.
- **Implement the function that creates and edits Max heap by receiving the [input].txt file as input as below, and the output result is stored in the [output].txt file.**
-



- <input> :
  각 line 마다 Command 가 주어짐.(**create, insert, delete, find, print**)
  Commands are given line by line (**create, insert, delete, find, print**)
  Key 값은 양의 정수
  Key values are **positive integers**

- <output> :
  각 **Command** 에 맞는 **result** 출력**.**
  Appropriate output messages for each command.
  모든 파일출력은 **main** 함수에서만 실행
  All file outputs are performed only in the main function.

<Create_New_MaxHeap>

- 위의 사진에서처럼 **"n [Heap의 capacity]"** 로 표현합니다.
- "n 10" 의 경우, capacity가 10인 MaxHeap을 생성하시면 됩니다.
- **"n [HeapCapacity]"** 로 표현합니다.
- In the case of "n 10", Max Heap with capacity of 10 should be generated.

<Insert>

- 위의 사진에서처럼 **"i [insert할 key값]"** 로 표현합니다.
- "i 1" 의 경우 "1" key를 MaxHeap에 insert 해주시면 됩니다.

- Heap 안에 이미 존재하는 Key를 insert 하는 경우나, Heap이 꽉 차있는 상태인 경우에 따라 에러 메시지를 출력해주셔야 합니다.
- **Insert** 함수는 **Insert** 성공**,** 실패에 따른 결과를 반환합니다**(Success: 0, Full: 1, Duplicated: 2).**
- key 중복시 에러메시지 형식은**"insert error : key값 is already in the heap"** 입니다.
- Heap이 꽉찬경우 에러메시지 형식은 **"insert error : heap is full"** 입니다.
- insertion 이 수행될 때마다 위 사진처럼 **"insert key값"** 을 출력하고, insertion 직후 Heap의 상태는 올바른 MaxHeap의 상태에 맞게 조정되어야 합니다.
- **"i [key value to be inserted]"**
- In the case of "i 1", "1" key should be inserted into Max Heap.
- Error messages should be printed when inserting the key value that already exists in the heap or Heap is full.
- **The Insert function returns a result based on whether the insertion is successful or not, and the main function performs file output based on the result**.
- When inserting the key value that already exists, the format of the error message is **"insert error : keyValue is already in the heap"**.
- When inserting the key value into full state Heap, the format of the error message is **"insert error : heap is full"**.
- **"insert keyValue"** should be printed out for every insertion, and the state of the heap should be properly arranged as appropriate state for Max heap.

<Delete>

- 위 사진처럼 **"d"** 로 표현합니다.
- "d" 의 경우, MaxHeap에서 Max Element를 제거해주셔야 합니다.
- Heap이 비어있는 상태에서 Deletion을 수행하는 경우 에러메시지를 출력해주셔야 합니다.
- **DeleteMax** 함수는 삭제에 성공하면 삭제한 값을 반환하고**, heap**이 비어있다면**, 0**을 반환하면 됩니다**.**
- 에러메시지 형식은 **"delete error : heap is empty"** 입니다.
- Deletion이 수행될 때마다 위 사진처럼 **"max element : key값 deleted"** 를 출력하고, Deletion 직후 Heap의 상태는 올바른 Maxheap의 상태에 맞게 조정되어야 합니다.
- **"d"**.
- In the case of "d", the Max element should be eliminated from the heap.
- Error message should be printed when deleting max from the empty heap, and the format of the error message is **"delete error : heap is empty"**.
- **The DeleteMax function should return the value that was deleted if the deletion is successful, or return 0 if the heap is empty.**
- **"max element : keyValue deleted"** should be printed out for every deletion, and the state of the heap should be properly arranged as appropriate state for Max heap.

<Find>

- 위의 사진처럼 **"f [search할 key값]"** 로 표현합니다.
- "f 2" 의 경우, Heap안에 "2"가 존재하는지 존재하지 않는지 출력해주시면 됩니다.
- **Find** 함수는 **key**값을 찾는데 성공하면 **1,** 실패하면 **0**을 반환하면 됩니다**.**
- 출력 형식은 key 값이 존재할 경우 **"key값 is in the heap"**
  key 값이 존재하지 않는 경우 **"finding error : key값 is not in the heap"** 입니다.
- **"f [key value to be searched]"**.
- In the case of "f 2", the existence of the key value in the Heap should be printed out.
- **The Find function should return 1 if the key value is found, and 0 if it fails to find the key value.**
- When the key value exists in the heap, **"keyValue is in the heap",**
  when the key value does not exist in the heap, **"finding error : keyValue is not in the heap"** should be printed out.

<getElements>

- 위 사진처럼 **"p"** 로 표현 합니다.

- "p"의 경우, Level Order 로 heap 안에 있는 Key 값들을 출력해주셔야 합니다.(root node 부터 index 순서대로 출력해주시면 됩니다.)
- **getElements** 함수는 **heap size** 크기의 **array**를 할당하여 **heap** 내부의 **Key**값을 **Level Order**로 복사한 후 그 **array**를 반환합니다**.**
- **main** 함수에서 **getElements** 함수로 부터 얻은 **array**에 저장된 **Key**값을 출력하면 됩니다**.**
- 출력후 **array**는 메모리 해제를 수행할 것이기 때문에 **heap** 구조체에 저장된 **Elements**의 포인터를 반환하면 안됩니다**.**
- Heap이 비어있는 경우 에러 메시지를 출력해주셔야 합니다.
- 에러메시지 형식은 **“print error : heap is empty”**입니다.
- **"p"**
- In the case of "p", all key values in the heap should be printed out in Level Order(in index order from the root node).
- **The getElements function allocates an array of the size of the heap, copies the key values in the heap in level order to the array, and returns the array.**
- **In the main function, you should print the key values stored in the array obtained from the getElements function**
- **Do not simply return a pointer of Elements stored in the heap structure, as the array will be deallocated after being printed.**
- When printing an empty heap, the error message should be printed in the format of **“print error : heap is empty”**.

<Structure & Function Format>

Structure

```
typedef struct HeapStruct{
    int Capacity;
    int Size;
    int *Elements;
}Heap;
```

Function

```
Heap* CreateHeap(int heapSize);
int Insert(Heap *heap, int value);
int Find(Heap *heap, int value);
int DeleteMax(Heap* heap);
int* getElements(Heap* heap);
int IsFull(Heap *heap);
int IsEmpty(Heap *heap);
int Size(Heap *heap);
```

- 위 사진과 같은 Struct 구조체를 사용하셔야 합니다.
- 위 사진과 같은 함수들을 형식에 맞게 구현해주시면 됩니다.
- Struct format above should be used for implementation
- Functions should be implemented in appropriate format as above.
-

<File Name Format>

- [StudentID].c ex) 20XXXXXXXX.c

<Execution>

- gcc 20XXXXXXXX.c -o 20XXXXXXXX
- ./20XXXXXXXX [input_file_name] [output_file_name]
- !!! 꼭 제공되는 testcase로 실행시켜보시기 바랍니다.!!!!!,
- !!! Run your solution code with the provided test case above and check whether it works properly !!!
-
-

<Issue>

- 코드 작성시 주석을 적어주시기 바랍니다. 주석이 없는경우 Cheating으로 간주될 수 있습니다.
- 제공된 testcase는 채점 case에 포함됩니다. 모두 알맞게 나오는지 확인해보시기 바랍니다.
- 파일 입출력은 *argv[]* 를 사용하여 구현해주시기 바랍니다.
- 제출 마감 시간 이전의 가장 최신 버전의 commit을 기준으로 채점할 예정입니다.
- 제출 파일과, 폴더 naming 은 꼭 지정된 형식으로 해주셔야 합니다.
- Please write down the detailed comments when writing the code. If there is no comment, it might be considered cheating.
- Provided test case is included in the test case for grading. Please check to see if it makes a proper result.
- Do not use a fixed file name when inputting and outputting files, but implement it using argv[] as in skeleton code.
- Scoring will be based on the latest version of commit before the deadline.
- The names of the .c file and directory should be named in proper format.

- 출력시, 위 사진의 예시와 같은 형식으로 출력해주시면 됩니다. 모든 공백은 띄어쓰기 한칸입니다. 모든 출력 메시지의 알파벳은 소문자만 사용하여 출력합니다.
- -> 출력시 위 사진과 같이 커맨드에 알맞는 메시지 출력후 줄바꿈.
- All the messages must be printed out according to the appropriate format as shown in the example above. All spaces are one space. Only lowercase letters should be used for the alphabets in output messages.
- -> newline(\n) after each output message.

<Directory Format>

- 아래와 같이 git 프로젝트 폴더에 “lab08” 폴더 생성후, "lab08" 폴더 안에 "20XXXXXXXX.c" 파일을 위치시키시면 됩니다.
- After creating the "lab08" directory in the git-project-directory as below, place the "20XXXXXXXXXX.c" file in the "lab08" directory.

>>

2023_CSE2010_20XXXXXXXX/ (GitLab project directory)

---...

---lab06/

-------2022XXXXXX.c

—-----...

---lab07/

-------2022XXXXXX.c

—-----...

---lab08/
-------2022XXXXXX.c
>> QnA : 2023ds12282@gmail.com