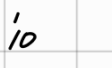
수업시간에 배운 토너먼트 방식으로 최대값을 구하는 방법에 배열이 사용될 경우 N개의 입력데이터에 대해 2N+1개의 공간이 필요하다. 공간을 절약하기 위해 N개의 공간만 필요한 변형된 토너먼트 방식을 구현하고자 한다.

수업시간에 배운 토너먼트와 마찬가지로 2진 트리 형태가 되며, 배열 1번, 즉 루트에 최대값을 갖는다. 이를 위해, 입력은 배열의 1번 인덱스부터 하나씩 증가하며 순차적으로 이루어지며, 1번에 최대값을 유지하기 위해 부모 노드와 비교하여 현재 입력된 데이터가 그 부모보다 크다면 부모와 자리 바꿈을 하고 루트까지 이를 반복한다.

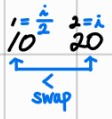
인덱스 i의 부모 : $\frac{i}{2}$

입력 : 10, 20, 5, 30, 15, 40

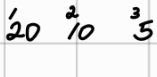
1. 입력 10
   결과 [10]



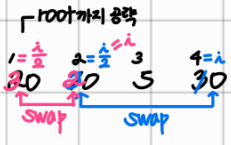
2. 입력 20
   결과 [20, 10]



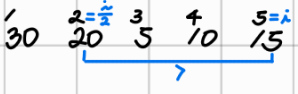
3. 입력 5
   결과 [20, 10, 5]



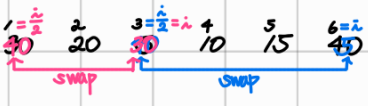
4. 입력 : 30
   결과 [30, 20, 5, 10]

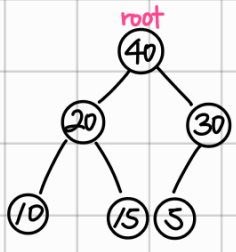


5. 입력 : 15
   결과 [30, 20, 5, 10, 15]



6. 입력 : 40
   결과 [40, 20, 30, 10, 15, 5]





요구사항

토너먼트 배열은 makeTournament 함수에서 만들어지고, 배열은 main에서 선언되어 매개변수로 makeTournament 함수로 전달되며, tournament 배열의 실제 크기를 확인하여 출력하여야 한다.

```
int main(){
    int tournamentArray[200];
    makeTournament(tournamentArray);
}
```

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

void makeTournament(int tournamentArray[]);
int swap(int*, int*);
void

int main(){
    int tournamentArray[200];
    makeTournament(tournamentArray);
}

void makeTournament(int tournamentArray[]){
    int index = 1; // index 1부터 시작
    int num; // 배열에 입력할 값
    while(1){
        printf("n번 배열에 입력할 값을 입력하시오 : ");
        scanf("%d",&num);
        tournamentArray[index] = num;
        int i = index;
        while(i/2 >= 1 && tournamentArray[i/2] < tournamentArray[i]){
            swap(&tournamentArray[i/2], &tournamentArray[i]);
            i = i/2; // root까지 비교하기 위해 i 재설정
        }
        for(int i = 1; i <= index; i++){ // 배열 출력
            printf("[%d] ",tournamentArray[i]);
        }
        index++;
    }
}
```

```c
int swap(int* x, int* y){
void
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
    (return temp;) 없어도 됨(int → void)
}
```

맨 처음 입력 시, 0=i/2, 1=i 째에 10 부모 노드 쓰레기값 출력됨. 루트는 인덱스 1이므로, 부모 노드의 인덱스는 1보다 크거나 같음

index = index/2가 되면 index의 값이 바뀌므로, index 값은 fix하고 root까지 비교하기 위해 변수 i에 index값 대입

1=i/2 2=i/2 3=i 4=i
10 20 5 30
swap swap