

BAEKJOON - 15663

# N과 M (9)

2 0 2 0 2 1 0 2 0    우 나 현

## 15663 - N과 M (9)

### 문제

#### N과 M (9) 성공

2 실버 II

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	512 MB	13058	6661	4768	49.641%

#### 문제

N개의 자연수와 자연수 M이 주어졌을 때, 아래 조건을 만족하는 길이가 M인 수열을 모두 구하는 프로그램을 작성하시오.

- N개의 자연수 중에서 M개를 고른 수열

#### 입력

첫째 줄에 N과 M이 주어진다. ( $1 \leq M \leq N \leq 8$ )

둘째 줄에 N개의 수가 주어진다. 입력으로 주어지는 수는 10,000보다 작거나 같은 자연수이다.

#### 출력

한 줄에 하나씩 문제의 조건을 만족하는 수열을 출력한다. 중복되는 수열을 여러 번 출력하면 안되며, 각 수열은 공백으로 구분해서 출력해야 한다.

수열은 사전 순으로 증가하는 순서로 출력해야 한다.

15663 - N과 M (9)

## 문제

예를 들어,

N이 4 , M이 2

N이 차례대로 9 7 1 9 라고 입력되었을 때

1 7

1 9

7 1

7 9

9 1

9 7

9 9 이런 식으로 출력되는 수열.

15663 - N과 M (9)

접근

9 가 두개 입력되어 중복 되지만

1 7

1 9

1 9

이런식으로 두개가 나오지 않고 하나만 적용되어야 한다.

재귀함수를 사용

15663 - N과 M (9)

## 코드

```
#include<iostream>
#include<algorithm>
using namespace std;
int N, M;
int* num = new int[N + 1]();
int arr[9];
bool visited[9];
```

수열은 오름차순으로 출력되어야 하므로,  
Algorithm 라이브러리를 호출해줌.

Num은 동적할당으로 수열이 0이 아닌  
1부터 시작하기 때문에 인덱스 0을 비워줘야  
하기 때문에 N + 1 크기의 배열을  
만들어 준다.

Arr은 수열을 저장하는 위치  
Visited는 1 1 과 같이 중복을 제외 하기 위해  
방문 했는지 안 했는지 를 체크.

15663 - N과 M (9)

## 코드

```
int main() {  
    cin >> N >> M;  
    for (int i = 1; i < N + 1; i++)  
        cin >> num[i];  
    sort(num, num + N + 1);  
    func(0);  
    return 0;  
}
```

## 메인

N과 M을 입력 받고  
N번 만큼 반복문을 돌려서 수를 입력 받는다.

Sort함수를 사용하여 오름차순 정렬.

Func는 함수. 0을 대입해준다.

15663 - N과 M (9)

## 코드

```
void func(int cnt) {
    if (cnt == M) {
        for (int i = 0; i < M; i++)
            cout << arr[i] << " ";
        cout << "\n";
        return;
    }

    int before = 0;
    for (int i = 1; i <= N; i++) {
        if (!visited[i] && before != num[i]) {
            visited[i] = true;
            arr[cnt] = num[i];
            before = num[i];
            func(cnt + 1);
            visited[i] = false;
        }
    }
}
```

## 수열을 계산해 주는 함수

우선 변수 cnt를 0으로 받아온다.

그리고 cnt가 출력 해야 하는 수열의 개수인 M과 같아지면 수열을 차례대로 출력.

Before은 중복을 잡아 내기 위한 변수.

입력 받은 N만큼 반복문을 돌려 주는데, 만일 visited가 false이면서 입력 받아져 있는 num[i]와 전에 방문 했던 수가 같지 않으면 조건문으로 들어 감.

## 15663 - N과 M (9)

### 코드

```
void func(int cnt) {  
    if (cnt == M) {  
        for (int i = 0; i < M; i++)  
            cout << arr[i] << " ";  
        cout << "\n";  
        return;  
    }  
    int before = 0;  
    for (int i = 1; i <= N; i++) {  
        if (!visited[i] && before != num[i]) {  
            visited[i] = true;  
            arr[cnt] = num[i];  
            before = num[i];  
            func(cnt + 1);  
            visited[i] = false;  
        }  
    }  
}
```

그리고 방문한 배열은 true로 바꿔주고  
수열을 저장 할 배열  
Arr에 num을 입력해준 후  
Before에도 num을 입력 해 준다.

그리고 다시 func를 호출.  
첫번째 조건문에 해당하면 출력

그리고 다시 false로 바꿔준다.



15663 - N과 M (9)

## 코드

### 코드 실행 예시)

N = 3 , M = 1

Num = 4 4 2 를 입력 후

Sort로 정렬.

-> Num { 0 , 2 , 4 , 4 }

(func 함수 호출)Func(0) -> Cnt = 0

아직 Cnt는 0이므로 조건문 제외.

Before =0 으로 초기화.

Visited {false, false, false ...}

첫번째로 들어갈 인덱스는 1부터 시작.

False이고 num[1] = 2

Visited{ false, true, false ...}

Arr { 2, 0 , 0 ... }

Before = 2

그리고 다시 func함수 호출 (재귀)

Func(Cnt +1)

이므로 Cnt = 1

15663 - N과 M (9)

## 코드

그럼  $M == \text{Cnt}$  가 되므로  
2를 출력.

그리고 return 으로 다시 돌아가,  
Visited {false, false, false...}

이제 다시 for문으로 돌아가  
 $i = 2$

이런식으로 for문이 다 돌면 끝이남.

감 사 합 니 다