

# C프로그래밍 (CSE2035)

## (실습 4)

---

**Ji-Hwan Kim, Ph.D.**

**Dept. of Computer Science and Engineering**

**Sogang University**

**Seoul, Korea**

**Tel: +82-2-705-8924**

**Email : [kimjihwan@sogang.ac.kr](mailto:kimjihwan@sogang.ac.kr)**



## 제출 형식

---

1. 각 문제에 대한 소스 코드를 압축하여 사이버캠퍼스에 업로드
  - 압축 파일명: "[실습#]학번\_이름.zip" (#은 실습번호)
  - 각 소스코드 파일명: "cp실습번호\_학번\_p문제번호.c"

1. 제출 기한 내에 사이버캠퍼스에 제출
  - 화요일에 업로드 된 경우: 이번주 수요일 오후 11시 59분까지
  - 목요일에 업로드 된 경우: 이번주 금요일 오후 11시 59분까지

2. COPY 등의 문제 발생 시 실습 0점 및 각종 불이익을 줄 것



## Practice 1.

1~n 까지의 서로 다른 정수 n개를 배열 a에 입력 받고  $i(0 \leq i < n)$ 에 대하여  $a[i] \neq b[i]$ 인 1부터 n까지의 서로 다른 n개의 수로 이루어진 배열 b를 모두 출력하시오.

입력 : 첫째 줄에 정수 n을 입력받는다. ( $2 \leq n \leq 8$ )

둘째 줄에 1부터 n까지의 서로 다른 정수 n개를 입력받는다.

출력 : 첫째 줄에 배열 b를 출력한다.

### 제약조건

- [] 연산자는 배열 선언할 때만 사용할 수 있다.
- 전역변수는 사용할 수 없다.



# Practice 1.

```
gyujin@kresnik:~/c_prog$ ./4_1
4
4 3 2 1
1 2 3 4
1 2 4 3
1 4 3 2
2 1 3 4
2 1 4 3
2 4 1 3
3 1 4 2
3 2 1 4
3 4 1 2
gyujin@kresnik:~/c_prog$ ./4_1
3
3 2 1
1 3 2
2 1 3
gyujin@kresnik:~/c_prog$ ./4_1
3
1 2 3
2 3 1
3 1 2
gyujin@kresnik:~/c_prog$ ./4_1
3
2 1 3
1 3 2
3 2 1
gyujin@kresnik:~/c_prog$ ./4_1
2
1 2
2 1
```



## Practice 2.

N행 M열의 2차원 행렬과 K가 입력으로 주어질 때, 이분탐색을 이용하여 주어진 행렬에 K가 존재하면 1과 찾은 값의 주소를, 존재하지 않으면 0을 출력하라

(\*\* 이분탐색은 정렬을 반드시 필요로 한다. \*\*)

입력: 첫째 줄에 정수 N,M,K가 주어진다. ( $1 \leq N, M \leq 100$ ) 두번째 줄부터 N+1번째 줄까지 각 줄마다 M개의 정수가 주어진다.

출력: 첫째 줄에 1 혹은 0을 출력한다. 만약 찾았다면 두번째 줄에 값의 주소를 출력한다.(여러 개의 답이 존재 한다면 하나만 출력한다.)

### 제약조건

- 전역변수는 사용할 수 없다.
- 다음장의 함수를 전부 구현해야 한다.



## Practice 2.

### 작성해야 하는 함수

- **void swap(int \*a,int \*b)**
  - 입력받은 두 수를 Swap 하는 함수
- **void sort(int \*a,int m)**
  - m개의 1차원 배열을 오름차순 정렬하는 함수이다.
- **void all\_sort(int a[][100],int n,int m)**
  - n행 m열의 2차원 배열을 행별로 정렬하는 함수이다.
- **int binary\_search(int \*a,int\* endPtr,int key,int \*\*findPtr)**
  - m개의 1차원 배열에서 key가 있으면 1을, 없으면 0을 반환한다.
  - findPtr에 찾은 값의 주소를 저장한다.
- **int all\_binary\_search(int a[][100],int n,int m,int key,int \*\*findPtr)**
  - n행 m열의 2차원 배열에서 key가 있으면 1을, 없으면 0을 반환한다.



## Practice 2.

### *Example*

```
gr120160213@cspro:~/exer_test$ ./a.out
3 3 1
3 3 3
3 3 1
3 3 3
1
0x7fffecdd2f0c
gr120160213@cspro:~/exer_test$ ./a.out
2 3 11
9 8 5
11 11 2
1
0x7fff5ab25510
```



## Practice 3.

$N \times M$  크기의 보드판이 있다. 보드판의 모든 격자(총  $N \times M$ 개가 있다)에는 높이가 2인 막대가 있다. 막대는 오른쪽 또는 아래로 넘어트릴 수 있다. 현재 격자의 위치가  $(y, x)$ 라고 할 때 오른쪽으로 넘어트리면 막대는  $(y, x)$ 와  $(y, x+1)$ 의 격자 위에 덮어지게 된다. 격자 위에 막대가 덮여져 있으면 그 격자에서는 막대를 넘어트릴 수 없다.

그리고 입력으로는 막대를 넘어트리고 싶은 방향에 대한 정보가 들어오게 되는데 0일 때는 오른쪽으로 덮고, 1일 때는 아래로 덮으려고 한다. 그러나 넘어트릴 수 없는 경우가 생길 수 있는데 그 때는 다른 방향으로 넘어트리면 된다. 그러나 다른 방향으로도 넘어트릴 수 없으면 다음 격자로 넘어가면 된다.

또한 막대를 넘어트릴 때 막대의 끝이 보드판을 벗어나게 되면 안된다.

그리고 막대를 넘어트리는 방향으로 위에서 아래로, 왼쪽에서 오른쪽으로 진행한다. 즉,  $(0, 0) \rightarrow (0, 1) \rightarrow \dots \rightarrow (0, m-1) \rightarrow (1, 0) \rightarrow \dots \rightarrow (n-1, m-1)$  순으로 진행한다.





## Practice 3.

---

입력: 첫째 줄에 정수  $N$ 과  $M$ 을 입력한다.(  $1 \leq N, M \leq 100$  ) 두번째 줄부터  $N+1$ 번째 줄까지 각 줄마다  $M$ 개의 정수를 입력한다.

출력: 옳어진 막대 수를 출력

### 제약조건

- 전역변수는 사용할 수 없다.
- `void count(int board[][100],int n,int m)`로 함수를 선언해서 구현할 것. ( $n$ 은 행의 개수,  $m$ 은 열의 개수이다)
- 배열의 선언과 함수의 선언을 제외하고 [] 연산자를 쓸 수 없다.

# Practice 3.

## Example

```
gyujin@kresnik:~/c_prog$ ./4_3
3 4
1 0 0 0
0 1 1 1
0 0 1 1
5
```

```
1223
1453
_45_
_
```

막대를 조건에 맞춰서 엮어트리게 되면 옆의  
그림과 같이 엮어지게 된다.  
\_으로 되어 있는 부분은 막대가 엮어져 있지  
않는 부분이다.