

C프로그래밍 (CSE2035) (실습3)

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



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.

1부터 10까지 저장되어있는 정수형 배열을 n 번 오른쪽으로 이동하거나 n 번 왼쪽으로 이동하는 프로그램을 작성하시오. 다음 세 가지 함수를 작성해야한다.

void shift_n_left(int *a,int n) : a 배열을 왼쪽으로 n 번 이동시킨다.

void shift_n_right(int *a,int n) : a 배열을 오른쪽으로 n 번 이동시킨다.

void print_array(int *a) : 배열 a를 출력한다.

입력 : 첫째 줄에 정수 n 을 입력받는다. ($0 \leq n \leq 10$)

둘째 줄에는 왼쪽으로 이동시킬지, 오른쪽으로 이동시킬지를 나타내는 숫자를 입력받는다. (0이면 왼쪽으로, 1이면 오른쪽으로)

출력 : 첫째 줄에 배열 a를 띄어쓰기로 구분하여 출력한다.

제약조건

- main 함수에서 `int a[10]={1,2,3,4,5,6,7,8,9,10};` 을 선언한다.
- [] 연산자는 배열 선언할 때만 사용할 수 있다.
- 전역변수는 사용할 수 없다.



Practice 1.

Example

```
gyujin@kresnik:~/c_prog$ ./3_2
3
0
4 5 6 7 8 9 10 1 2 3
gyujin@kresnik:~/c_prog$ ./3_2
3
1
8 9 10 1 2 3 4 5 6 7
gyujin@kresnik:~/c_prog$ ./3_2
5
1
6 7 8 9 10 1 2 3 4 5
gyujin@kresnik:~/c_prog$ ./3_2
5
0
6 7 8 9 10 1 2 3 4 5
gyujin@kresnik:~/c_prog$ ./3_2
4
0
5 6 7 8 9 10 1 2 3 4
gyujin@kresnik:~/c_prog$ ./3_2
6
1
5 6 7 8 9 10 1 2 3 4
gyujin@kresnik:~/c_prog$
```



Practice 2.

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 만들고 출력하시오. 다음 두 가지 함수를 작성해야 한다.

void make_array(int *a, int *b, int n) : $a[i] \neq b[i]$ 를 만족하는 크기가 n인 배열 b를 만든다. (n은 a의 배열의 크기와 같다) 필요시, 함수에 매개변수를 추가해도 좋다.

Void print_array(int *a, int n) : 크기가 n인 정수배열 a를 출력한다.

입력 : 첫째 줄에 정수 n을 입력받는다. ($0 \leq n \leq 10$)

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.

배열 b는 문제에서 주어진 조건만 만족하는 어떤 배열이면 된다.



Practice 2.

Example

```
gyujin@kresnik:~/c_prog$ ./3_1
4
3 2 1 4
1 3 4 2
gyujin@kresnik:~/c_prog$ ./3_1
5
4 3 2 1 5
1 2 3 5 4
gyujin@kresnik:~/c_prog$ ./3_1
5
1 2 3 4 5
2 1 4 5 3
gyujin@kresnik:~/c_prog$ ./3_1
4
1 3 2 4
2 1 4 3
gyujin@kresnik:~/c_prog$ ./3_1
6
1 2 6 5 4 3
2 1 3 4 5 6
```



Practice 3.

$N \times N$ 크기의 두 행렬을 입력받아 두 행렬의 합 또는 차 또는 곱을 출력한다.

입력: 첫째 줄에 정수 N ($1 \leq N \leq 100$)이 주어진다. 그리고 다음 둘째 줄부터 $N+1$ 번째까지 첫번째 행렬이, $N+2$ 번째 줄부터 $2N+1$ 번째 줄까지 두번째 행렬이 주어진다.

둘째 줄에는 합을 구할지, 차를 구할지, 곱을 구할지를 나타내는 숫자가 주어진다. (0이면 합, 1이면 차, 2이면 곱)

출력: 첫째 줄부터 N 번째 줄까지 두 행렬 곱을 출력한다.

제약조건

- `void func(int a[100][100], b[100][100], c[100][100], int N, int op)` 함수를 작성하고 사용하십시오.
 - 두 행렬을 입력받아 `op`에 따라 `c`행렬에 연산 결과를 저장한다.
- 전역변수는 사용할 수 없다.
- 배열의 선언과 함수의 선언을 제외하고 [] 연산자를 쓸 수 없다.

Practice 3.

행렬(matrix)이란?

수나 식을 직사각형 모양으로 배열한 것.

$$\begin{array}{c}
 \xrightarrow{\text{행(row)}} \\
 A = \begin{bmatrix} 3 & 0 & 5 \\ 2 & -3 & 1 \end{bmatrix}
 \end{array}
 \quad
 \begin{array}{c}
 B = \begin{bmatrix} 2 & 0 \\ 3 & 7 \\ -1 & 0 \end{bmatrix} \\
 \downarrow \text{열(column)}
 \end{array}$$

행렬의 덧셈과 뺄셈

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$$

$$A + B = \begin{bmatrix} 1+4 & 2+3 \\ 3+2 & 4+1 \end{bmatrix} = \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix}$$

$$A - B = \begin{bmatrix} 1-4 & 2-3 \\ 3-2 & 4-1 \end{bmatrix} = \begin{bmatrix} -3 & -1 \\ 1 & 3 \end{bmatrix}$$

Practice 3.

행렬의 곱셈

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$$

$$A * B = \begin{bmatrix} 1 * 4 + 2 * 2 & 1 * 3 + 2 * 1 \\ 3 * 4 + 4 * 2 & 3 * 3 + 4 * 1 \end{bmatrix} = \begin{bmatrix} 8 & 5 \\ 20 & 13 \end{bmatrix}$$

$$A = \begin{bmatrix} 3 & 0 & 5 \\ 2 & -3 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 0 \\ 3 & 7 \\ -1 & 0 \end{bmatrix}$$

$$A * B = \begin{bmatrix} 3 * 2 + 0 * 3 + 5 * (-1) & 3 * 0 + 0 * 7 + 5 * 0 \\ 2 * 2 + (-3) * 3 + 1 * (-1) & 2 * 0 + (-3) * 7 + 1 * 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -6 & -21 \end{bmatrix}$$

Practice 3.

Example

```
gyujin@kresnik:~/c_prog$ ./3_3
3
1 2 0
0 1 1
2 0 1
1 1 2
2 1 1
1 2 1
2
5 3 4
3 3 2
3 4 5
gyujin@kresnik:~/c_prog$ ./3_3
3
1 2 0
0 1 1
2 0 1
1 1 2
2 1 1
1 2 1
0
2 3 2
2 2 2
3 2 2
gyujin@kresnik:~/c_prog$ ./3_3
3
1 2 0
0 1 1
2 0 1
1 1 2
2 1 1
1 2 1
1
0 1 -2
-2 0 0
1 -2 0
gyujin@kresnik:~/c_prog$ ./3_3
2
3 0
0 3
4 0
0 4
2
12 0
0 12
```