



과제 05

- ❑ 뒷장에 주어지는 문제를 풀어, **이면지나 양면지**를 이용하여 샤프나 연필을 사용하여 **손으로** 적어서 **다음 주 시험 시작 전에** 제출합니다.
 - 문제에서 “함수를 작성하시오” 라고 되어 있는 경우에는, 지정한 함수만 적어서 내면 됩니다. 메인 함수까지 다 적지 않아도 됩니다.
 - 따라서, 이번 과제는 쓸 게 그다지 많지 않으므로, printer 출력을 허용하지 않습니다.
- ❑ 결과 예시에서 **붉은 글자**는 사용자가 입력하는 값을 의미합니다.
 - “*”나 “-”개수까지 정확하게 맞추지는 않아도 됩니다.
- ❑ 아직까지 배우지 않는 내용을 활용하지 마세요.
 - string class 등을 사용하지 마세요.
- ❑ 프로그램을 만일 하나의 파일로 (시험 환경처럼) 짤다면, 한 프로그램에 inputArray를 여러 개 overload 해서 만들 수 있습니다. 한번 해 보세요. ^^



05-1. 배열 입/출력 프로그램

- ❑ 아래의 왼쪽의 프로그램을 완성시켜서 오른쪽과 같이 작동되게 만드는, `inputArray`와 `printReverse` 함수를 작성하시오.
 - 마침표가 입력되면 입력의 마지막으로 보고, `myStr`을 partially filled array로 사용하시오. 이를 위하여 `inputArray`에서 세번째 매개변수를 reference로 사용하시오.

```
const int MAX_STR = 100 ;
char myStr[MAX_STR];
int strLength ;

while (1)
{
    cout << "문자열입력: " ;
    inputArray(myStr, MAX_STR, strLength) ;

    if (strLength == 0)
        break ;

    cout << "거꾸로출력: " ;
    printReverse(myStr, strLength) ;
}
```

문자열입력: **abcdefaxb.**
거꾸로출력: bxafedcba

문자열입력: **12345.**
거꾸로출력: 54321

문자열입력: **.**
종료합니다.



05-2. 최대값, 최소값, 평균 구하기

- ❑ 아래의 왼쪽의 프로그램을 완성시켜서 오른쪽과 같이 작동되게 만드는, findMinMaxAvg함수를 작성하시오.
 - 0 이하의 숫자가 들어오면 입력의 마지막으로 취급하시오.

```
const int MAX_SIZE = 100 ;  
int arr[MAX_SIZE];  
int len, min, max ;  
double avg ;
```

```
while (1)
```

```
{
```

```
    cout << MAX_SIZE << "개 이하의 자연수 입력: " ;  
    inputArray(arr, MAX_SIZE, len) ;
```

```
    if (len == 0)  
        break ;
```

```
    findMinMaxAvg(arr, len, min, max, avg) ;
```

```
    cout << "최대값은" << max << ", 최소값은" << min << ", 평균은" << avg << endl <<endl ;
```

```
}
```

100개 이하의 자연수 입력 : 1 5 7 9 8 4 1 0

최대값은 9, 최소값은 1, 평균은 5

100개 이하의 자연수 입력 : 1 2 3 4 5 6 7 8 9 10 11 12 -1

최대값은 12, 최소값은 1, 평균은 6.5

100개 이하의 자연수 입력 : -1

종료합니다.



05-3. 모음/자음 개수 판별 프로그램

- 아래의 왼쪽의 프로그램을 완성시켜서 오른쪽과 같이 작동되게 만드는, checkCharCount 함수를 작성하시오.

```
const int MAX_STR = 100 ;
char myStr[MAX_STR];
int strLength, vowelCnt = 0, upperCnt = 0, lowerCnt = 0 ;

while (1)
{
    cout << "문자열입력: " ;
    inputArray(myStr, MAX_STR, strLength) ;

    if (strLength == 0)
        break ;

    checkCharCount(myStr, strLength, upperCnt, lowerCnt, vowelCnt) ;
    cout << "대문자개수는: " << upperCnt << endl ;
    cout << "소문자개수는: " << lowerCnt << endl ;
    cout << "모음의개수는: " << vowelCnt << endl << endl ;
}
```

문자열 입력 : asdfas d;fja sdfasdfAS LDFKJA.

대문자 개수는 : 8

소문자 개수는 : 17

모음의 개수는 : 6

문자열 입력 : a sdf las ASDLF.

대문자 개수는 : 5

소문자 개수는 : 7

모음의 개수는 : 3

문자열 입력 : .

종료합니다.

05-4. ww 또는 wcw 형태의 문자열 확인

- “abab” 또는 “abceabc” 등은 문자열의 절반과 나머지 절반이 동일한 형태이다. 입력되는 문자열이 이러한 요건을 만족하는지 확인하는 함수 isWWstr 를 작성하여 아래와 같이 작동하도록 하시오.

```
const int MAX_STR = 100 ;
char myStr[MAX_STR] ;
int strLength ;
bool isWWFlag ;

while (1)
{
    cout << "문자열입력: " ;
    inputArray(myStr, MAX_STR, strLength) ;

    if (strLength == 0)
        break ;

    isWWFlag = isWWStr(myStr, strLength) ;
    if (isWWFlag)
        cout << "WW 또는 WcW 형태를만족합니다." << endl << endl ;
    else
        cout << "WW 또는 WcW 형태를만족하지않습니다." << endl << endl ;
}
```

문자열 입력 : **abaab.**
WW 또는 WcW 형태를 만족합니다.

문자열 입력 : **abaab.**
WW 또는 WcW 형태를 만족합니다.

문자열 입력 : **abccba.**
WW 또는 WcW 형태를 만족하지 않습니다.

문자열 입력 : **abcabcabc.**
WW 또는 WcW 형태를 만족하지 않습니다.

문자열 입력 : **0.**
WW 또는 WcW 형태를 만족합니다.

문자열 입력 : **0a00a.**
WW 또는 WcW 형태를 만족합니다.

문자열 입력 : **.**
종료합니다.



05-5. 2차원 배열에 값 채우기

- ❑ 배열의 행과 열 개수를 사용자에게 입력받고, 배열의 요소를 입력받아 출력하는 아래 프로그램을 완성하시오. 배열의 최대 행과 열의 개수는 100으로 한다.
- getRowColumn 에서는 범위에 넘어서는 값이 들어오면 다시 입력받게 하고, row와 column은 reference를 사용하여 결과값을 되돌려 받게 한다.
- printArray에서는 cout.width()를 이용하여 입력되는 값이 최대 5자리라고 가정하고, 아래와 같이 배열을 깔끔하게 찍어보시오.

```
const int SIZE = 100 ;  
int arr[SIZE][SIZE], row, column ;  
  
while (1)  
{  
    getRowColumn(SIZE, row, column) ;  
    if (row == 0)  
        break ;  
  
    inputArray(arr, row, column) ;  
    printArray(arr, row, column) ;  
}
```

```
행 개수를 입력하세요 (0~100사이. 0은 종료) : 120  
행 개수를 입력하세요 (0~100사이. 0은 종료) : 4  
열 개수를 입력하세요 (1~100 사이): 0  
열 개수를 입력하세요 (1~100 사이): 3  
12개의 값을 입력하세요 : 12 54 9 5 846 87 13 54  
87 51 54 6  
| 12 54 9|  
| 5 846 87|  
| 13 54 87|  
| 51 54 6|  
행 개수를 입력하세요 (0~100사이. 0은 종료) : 2  
열 개수를 입력하세요 (1~100 사이): 3  
6개의 값을 입력하세요 : 56 98 78 165 6517 4  
| 56 98 78|  
| 165 6517 4|  
행 개수를 입력하세요 (0~100사이. 0은 종료) : 0  
종료합니다.
```



05-6. 2차원 정방행렬 합 구하기

- 6에서 만든 inputArray와 printArray를 활용하여, 아래 왼쪽과 같이 두개의 행렬의 합을 구하는 getMatrixAdd 함수를 작성하시오.

```
const int SIZE = 100 ;
int matrix1[SIZE][SIZE], matrix2[SIZE][SIZE];
int resMatrix[SIZE][SIZE] ;
int row, column ;

while (1)
{
    getRowColumn(SIZE, row, column) ;
    if (row == 0)
        break ;

    cout << "첫 번째 행렬값 입력" << endl ;
    inputArray(matrix1, row, column) ;

    cout << "두 번째 행렬값 입력" << endl ;
    inputArray(matrix2, row, column) ;

    getMatrixAdd(matrix1, matrix2, resMatrix, row, column) ;

    cout << "행렬합 결과" << endl ;
    printArray(resMatrix, row, column) ;
}
```

행 개수를 입력하세요 (0~100사이. 0은 종료) : 3

열 개수를 입력하세요 (1~100 사이): 2

첫 번째 행렬 값 입력

6개의 값을 입력하세요 : 1 2 3 4 5 6

두 번째 행렬 값 입력

6개의 값을 입력하세요 : 6 5 4 3 2 1

행렬 합 결과

1	7	7
---	---	---

1	7	7
---	---	---

1	7	7
---	---	---

행 개수를 입력하세요 (0~100사이. 0은 종료) : 4

열 개수를 입력하세요 (1~100 사이): 3

첫 번째 행렬 값 입력

12개의 값을 입력하세요 : 1 2 3 4 5 6 7 8 9 10 11 12

두 번째 행렬 값 입력

12개의 값을 입력하세요 : 13 14 15 16 4 3 2 1 4 3 2 1

행렬 합 결과

1	14	16	18
---	----	----	----

1	20	9	9
---	----	---	---

1	9	9	13
---	---	---	----

1	13	13	13
---	----	----	----

행 개수를 입력하세요 (0~100사이. 0은 종료) : 0

종료합니다



05-7. 삼각행렬 구하기

- 몇 차 정방행렬을 만들지 사용자에게 입력받고, 해당 정방행렬의 요소를 입력받아 출력하고, 대각선 아래 성분을 모두 0으로 변경하고 그렇게 얻어진 삼각행렬을 출력하는 프로그램을 위한 함수 makeTrianglarArray를 작성하시오. 정방행렬의 최대 차수는 100으로 한다.

몇 차 행렬을 사용하시겠습니까? 4

16개의 값을 입력하세요 : 21 54 87 31 8 897 123 89

213 54 4 78 21 4 32 5

	21	54	87	31
	8	897	123	89
	213	54	4	78
	21	4	32	5

삼각행렬로 변화 후 :

	0	54	87	31
	0	0	123	89
	0	0	0	78
	0	0	0	0

```
const int SIZE = 100 ;
int matrix[SIZE][SIZE], size ;

while (1)
{
    cout << "몇 차 행렬을 사용하시겠습니까? " ;
    cin >> size ;

    if (size == 0)
        break ;

    inputArray(matrix, size) ;
    printArray(matrix, size) ;

    makeTrianglarArray(matrix, size) ;
    cout << endl << "삼각행렬로 변화 후: " << endl ;
    printArray(matrix, size) ;
}
```




05-8. 정방행렬 곱 구하기

- 곱을 구할 정방행렬이 몇 차인지 사용자에게 입력을 받아, 두 개의 정방행렬을 만들고, 정방행렬에 들어갈 값을 사용자에게 다시 입력받아, 두 행렬의 곱을 구하는 프로그램을 위한 함수 getMatrixMulti를 작성하시오.

```
const int MAX_SIZE = 100 ;
int matrix1[MAX_SIZE][MAX_SIZE], matrix2[MAX_SIZE][MAX_SIZE];
int resMatrix[MAX_SIZE][MAX_SIZE] ;
int size ;

while (1)
{
    cout << "몇 차 행렬의 곱을 구하시겠습니까? " ;
    cin >> size ;

    if (size == 0)
        break ;

    cout << "첫 번째 행렬 값 입력" << endl ;
    inputArray(matrix1, size) ;

    cout << "두 번째 행렬 값 입력" << endl ;
    inputArray(matrix2, size) ;

    getMatrixMulti(matrix1, matrix2, resMatrix, size) ;

    cout << "행렬 곱 결과" << endl ;
    printArray(resMatrix, size) ;
}
```

몇 차 행렬의 곱을 구하시겠습니까? 3

첫 번째 행렬 값 입력

9개의 값을 입력하세요 : 1 2 3 4 5 4 3 2 1

두 번째 행렬 값 입력

9개의 값을 입력하세요 : 1 2 1 2 1 1 1 3 4

행렬 곱 결과

| 8 13 15|

| 18 25 25|

| 8 11 9|

몇 차 행렬의 곱을 구하시겠습니까? 2

첫 번째 행렬 값 입력

4개의 값을 입력하세요 : 1 2 3 4

두 번째 행렬 값 입력

4개의 값을 입력하세요 : 4 3 2 1

행렬 곱 결과

| 8 5|

| 20 13|

몇 차 행렬의 곱을 구하시겠습니까? 0

종료합니다.



05-9. 글자 개수 세기

- ❑ 지난 실습 시험 마지막 문제를 함수를 이용하여 해결해보자. 사용자에게 최대 100글자의 문자열을 05-1 문제의 inputArray로 입력받고, 입력된 문자열에 포함된 각 영어 알파벳의 개수를 출력해야 한다. 대소문자 각각의 개수를 세는 함수 countChar와 결과를 출력하는 함수 printCharCnt를 1차원 배열 2개를 이용하는 방법과 2차원 배열 1개를 이용하는 방법의 두가지로 overload하여 작성하자.

```
const int CHAR_CNT = 26 ;
int bigCharCnt[CHAR_CNT] = {0}, smallCharCnt[CHAR_CNT] = {0};
int charCnt[CHAR_CNT][2] = {0} ; //[0]은대문자. [1]은소문자

const int MAX_STR = 100 ;
char myStr[MAX_STR] ;
int strLength ;

inputArray(myStr, MAX_STR, strLength) ;

cout << "1차원 배열 두 개 사용" << endl ;
countChar(myStr, strLength, bigCharCnt, smallCharCnt) ;
printCharCnt(bigCharCnt, smallCharCnt) ;

cout << endl << "2차원 배열 한 개 사용" << endl ;
countChar(myStr, strLength, charCnt) ;
printCharCnt(charCnt) ;
```

문장 입력 : as;dfjas DFJALSJ FASjf.

1차원 배열 두 개 사용

A의 개수는 2	a의 개수는 2
D의 개수는 1	d의 개수는 1
F의 개수는 2	f의 개수는 2
J의 개수는 2	j의 개수는 2
L의 개수는 1	l의 개수는 0
S의 개수는 2	s의 개수는 2

2차원 배열 한 개 사용

A의 개수는 2	a의 개수는 2
D의 개수는 1	d의 개수는 1
F의 개수는 2	f의 개수는 2
J의 개수는 2	j의 개수는 2
L의 개수는 1	l의 개수는 0
S의 개수는 2	s의 개수는 2