

HW1 배열을 사용하여 총합과 평균 구하기

```
01 #include <iostream>
02 using namespace std;
03 void main()
04 {
05     int a[5]={85, 90, 75, 100, 95};
06     int tot;
07     double avg;
```

**총합과 평균을 구하기 위한 숫자들의 개수를
배열의 크기를 얻어 사용하기
이 코드에서 5라는 상수를 사용하지 않기!**

```
15 cout << "Total = " << tot << "\n";
16 cout << "Average = " << avg << "\n";
17 }
```

HW2 배열을 사용하여 최대값 구하기

```
01 #include <iostream>
02 using namespace std;
03 void main()
04 {
05     int a[5];
06     int max;
07     int i;
08
09     cout << "Enter 5 integer numbers\n";
10
11     for(i=0; i<5; i++){
```

5개 숫자 cin으로 입력 받기

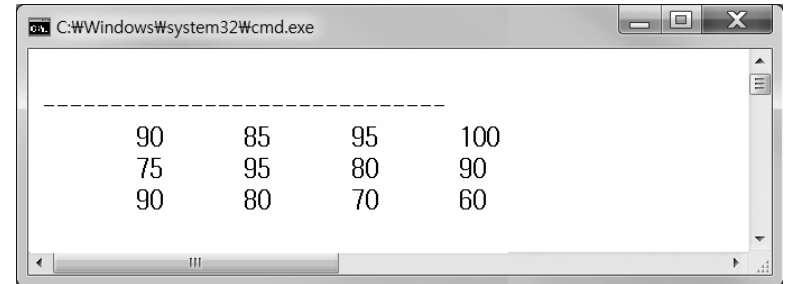
```
14 }
15
16 max = a[0];
```

처음 들어온 수를 max로 초기화
최종적으로 5가지 수 중 가장 큰 숫자가 max에
대입되도록

```
22 cout << "Maximum number : " << max << "\n";
23 }
```

HW3 2차원 배열 출력하기

```
01 #include <iostream>
02 using namespace std;
03 #define ROW 3
04 #define COL 4
05 void main()
06 {
07     int a[ROW][COL] = { {90, 85, 95, 100},
08 {75, 95, 80, 90},
09 {90, 80, 70, 60}
10 };
11
```



```
C:\Windows\system32\cmd.exe

-----
      90      85      95      100
      75      95      80      90
      90      80      70      60
```

```
21 }
```

HW4 두 행렬의 합 구하기

$$\begin{bmatrix} 10 & 20 & 30 & 40 \\ 20 & 40 & 60 & 80 \\ 10 & 30 & 50 & 70 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} =$$

1. `#define` 사용하여 2차원 행렬 크기 정의
2. 행렬의 값을 위와 같이 초기화 하기(대입이 아니라 초기화)
3. 행렬의 합을 아래와 같이 출력하기

```
11 22 33 44
25 46 67 88
19 40 61 82
```

HW5 rand() 함수로 주사위 6,000,000번 굴리는 시뮬레이션

```
#include <iostream>
#include <iomanip>
#include <cstdlib> // contains function prototype for rand
using namespace std;
```

```
int main()
{
```

```
    // summarize results of 6,000,000 rolls of a die
    for ( unsigned int roll = 1; roll <= 6000000; ++roll )
    {
```

//int face 선언.

//**rand()** 사용하여 주사위를 6,000,000굴려서

// 1~6 중 하나의 값을 무작위로 얻어서 face 변수에 대입

//face의 값에 따라 frequency의 값 하나씩 증가

//if 나 **switch** 사용하지 말고 size 6의 **frequency** 배열을 이용하여 각 face의 횟수 기록

Face	Frequency
1	999702
2	1000823
3	999378
4	998898
5	1000777
6	1000422

<= 지난 과제와 동일한 출력