

### 03. 개선된 합수 기능

## [실습 3-1]

- 함수 f()를 호출하는 경우가 다음과 같을 때 f()를 디폴트인자로 가진 함수로 작성하라.

```
f(); // 한 줄에 빈칸을 10개 출력한다.  
f('%'); // 한 줄에 '%'를 10개 출력한다.  
f('@', 5); // 다섯 줄에 '@'를 10개 출력한다.
```

```
#include <iostream>  
using namespace std;  
  
// 원형 선언  
void f(char c=' ', int line=1);
```

// 함수 구현

빈 칸이 10개  
출력됨

```
%%%%%%%%%%%%%  
@@@@@@@ @@@@  
@@@@@@@ @@@@  
@@@@@@@ @@@@  
@@@@@@@ @@@@  
@@@@@@@ @@@@  
@@@@@@@ @@@@
```

```
int main() {  
    f(); // 한 줄에 빈칸을 10개 출력한다.  
    f('%'); // 한 줄에 '%'를 10개 출력한다.  
    f('@', 5); // 5 줄에 '@' 문자를 10개 출력한다.  
}
```

## [실습 3-2]

- 함수 big()을 호출하는 경우는 다음과 같다.

```
int main() {  
    int x = big(3, 5); // 3과 5중 큰 값 5는 최대값 100보다 작으므로, 5 리턴  
    int y = big(300, 60); // 300과 60중 큰 값 300이 최대값 100보다 크므로, 100 리턴  
    int z = big(30, 60, 50); // 30과 60 중 큰 값 60이 최대값 50보다 크므로, 50 리턴  
    cout << x << ' ' << y << ' ' << z << endl;  
}
```

5 100 50

- (1) big() 함수를 2개 중복하여 작성하고 프로그램을 완성하라.
- (2) 디폴트 인자를 가진 하나의 함수로 big()을 작성하고 프로그램을 완성하라.

## [실습 3-3]

- 배열과 크기를 매개 변수로 받아 합을 구하여 리턴하는 제네릭 함수 add()를 작성하려고 한다. 빈칸을 채워 프로그램을 완성하시오.

```
#include <iostream>
using namespace std;
```

```
//제네릭 함수 add()
```

```
int main() {
    int x[] = {1,2,3,4,5};
    double d[] = {1.2, 2.3, 3.4, 4.5, 5.6, 6.7};
```

```
    cout << "sum of x[] = " << _____ << endl; // 배열 x와 원소 5개의 합을 계산
    cout << "sum of d[] = " << _____ << endl; // 배열 d와 원소 6개의 합을 계산
}
```

```
sum of x[] = 15
sum of d[] = 23.7
```

## [실습 3-4]

- 배열의 원소를 반대 순서로 뒤집는 reverseArray() 함수를 템플릿으로 작성하라. reverseArray()의 첫번째 매개변수는 배열에 대한 포인터이며, 두 번째 매개변수는 배열의 개수다. reverseArray() 호출 사례는 다음과 같다.

```
int main() {
    int x[] = {1, 10, 100, 5, 4};
    reverseArray(x, 5);
    for(int i=0; i<5; i++)
        cout << x[i] << ' '; // 4 5 100 10 1 이 출력된다.
    cout << endl;
}
```