

| 함수, 재귀



광운대학교
소프트웨어학부

김석희

2023.09.09

함수

A : 국어 5등급 수학 1등급 영어 6등급

B : 국어 2등급 수학 3등급 영어 2등급

C : 국어 4등급 수학 5등급 영어 1등급

D : 국어 1등급 수학 1등급 영어 1등급

학생들의 평균 모의고사 성적을 게시하려 함

함수

A : 국어 5등급 수학 1등급 영어 6등급
average 4

B : 국어 2등급 수학 3등급 영어 2등급
average 2.3

C : 국어 4등급 수학 5등급 영어 1등급
average 3.3

D : 국어 1등급 수학 1등급 영어 1등급
average 1

학생들의 평균 모의고사 성적을 게시하려 함

```
#include <iostream>

using namespace std;

int main() {

    int A_total = 5+1+6;
    double A_avrage = A_total / 3.0;
    cout << "A : " << A_avrage << " grade";

    int B_total = 2+3+2;
    double B_avrage = B_total / 3.0;
    cout << "B : " << B_avrage << " grade";

    int C_total = 4+5+1;
    double C_avrage = C_total / 3.0;
    cout << "C : " << C_avrage << " grade";

    int D_total = 1+1+1;
    double D_avrage = D_total / 3.0;
    cout << "D : " << D_avrage << " grade";
    return 0;
}
```

함수

A : 국어 5등급 수학 1등급 영어 6등급
average 4

B : 국어 2등급 수학 3등급 영어 2등급
average 2.3

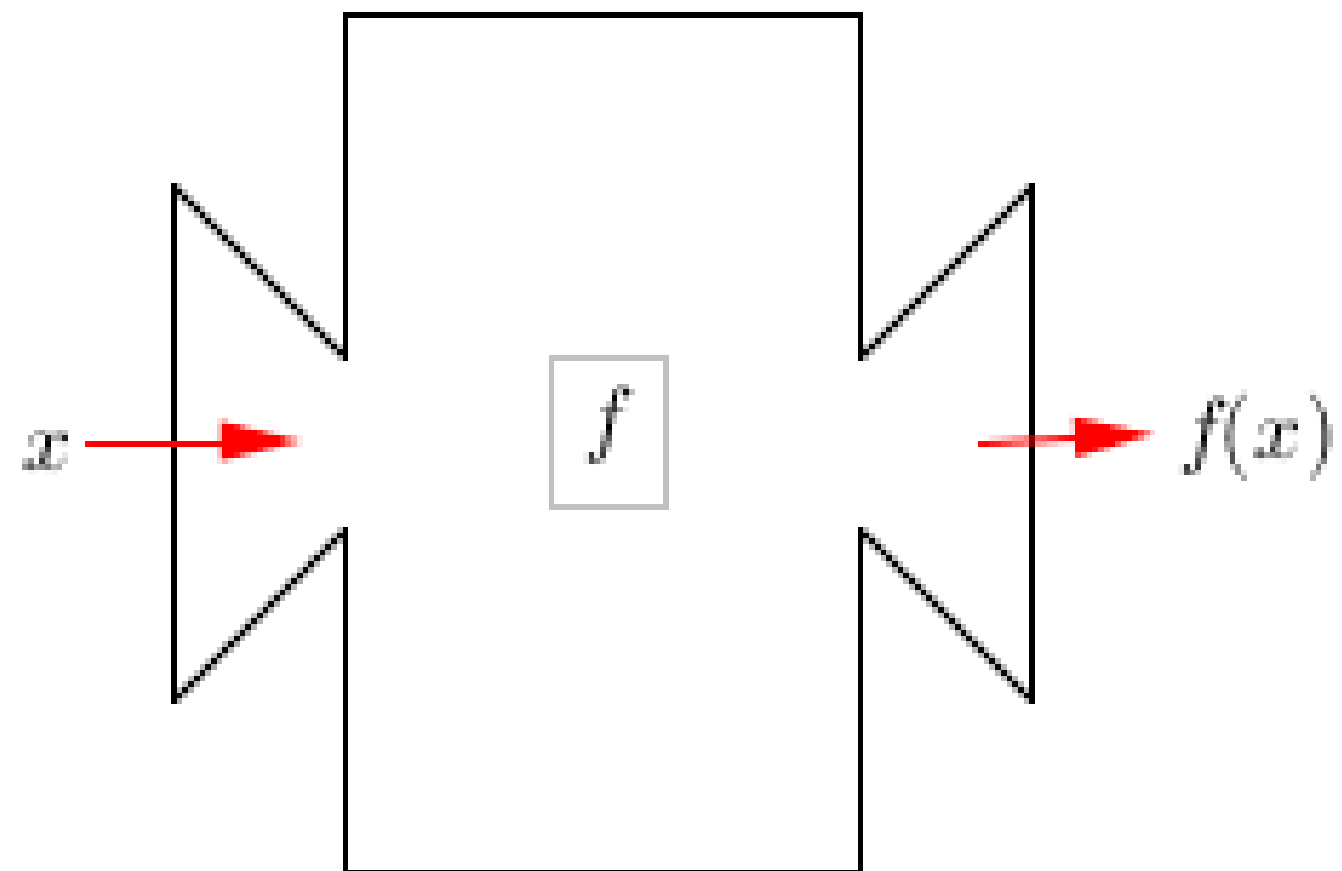
C : 국어 4등급 수학 5등급 영어 1등급
average 3.3

D : 국어 1등급 수학 1등급 영어 1등급
average 1

```
int A_total = 5+1+6;  
double A_avrage = A_total / 3.0;  
cout << "A : " << A_avrage << " grade";
```

학생들의 평균 모의고사 성적을 게시하려 함

함수



함수

```
#include <iostream>

using namespace std;

int main() {

    int A_total = 5+1+6;
    double A_avrage = A_total / 3.0;
    cout << "A : " << A_avrage << " grade";

    int B_total = 2+3+2;
    double B_avrage = B_total / 3.0;
    cout << "B : " << B_avrage << " grade";

    int C_total = 4+5+1;
    double C_avrage = C_total / 3.0;
    cout << "C : " << C_avrage << " grade";

    int D_total = 1+1+1;
    double D_avrage = D_total / 3.0;
    cout << "D : " << D_avrage << " grade";
    return 0;
}
```

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

void average(string name, int kor, int mat, int eng) {
    int total = kor + mat + eng;
    double avrage = total / 3.0;
    cout << name << " : " << avrage << " grade";
}

int main() {
    average("A", 5, 1, 6);
    average("B", 2, 3, 2);
    average("C", 4, 5, 1);
    average("D", 1, 1, 1);

    return 0;
}
```

함수

함수란?

프로그래밍에서 함수(function)란 하나의 특별한 목적의 작업을 수행하기 위해 독립적으로 설계된 프로그램 코드의 집합으로 정의할 수 있습니다. C 프로그램은 이러한 함수들로 구성되며, 포함된 함수들을 사용하여 프로그램의 목적을 달성하게 됩니다.

함수를 사용하는 이유

함수를 사용하는 가장 큰 이유는 바로 반복적인 프로그래밍을 피할 수 있기 때문입니다.

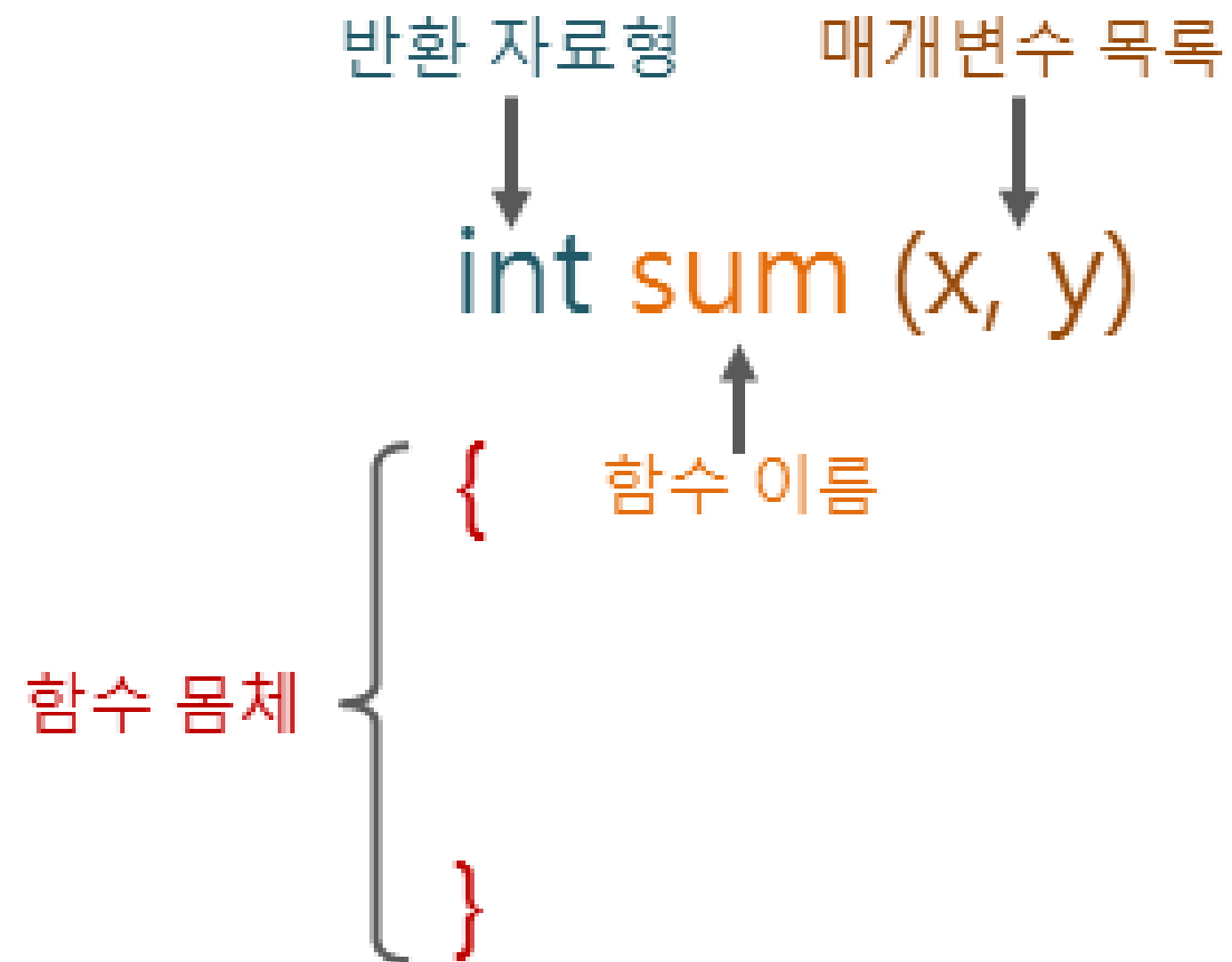
프로그램에서 특정 작업을 여러 번 반복해야 할 때는 해당 작업을 수행하는 함수를 작성하면 됩니다.

그리고서 프로그램이 필요할 때마다 작성한 함수를 호출하면 해당 작업을 반복해서 수행할 수 있습니다.

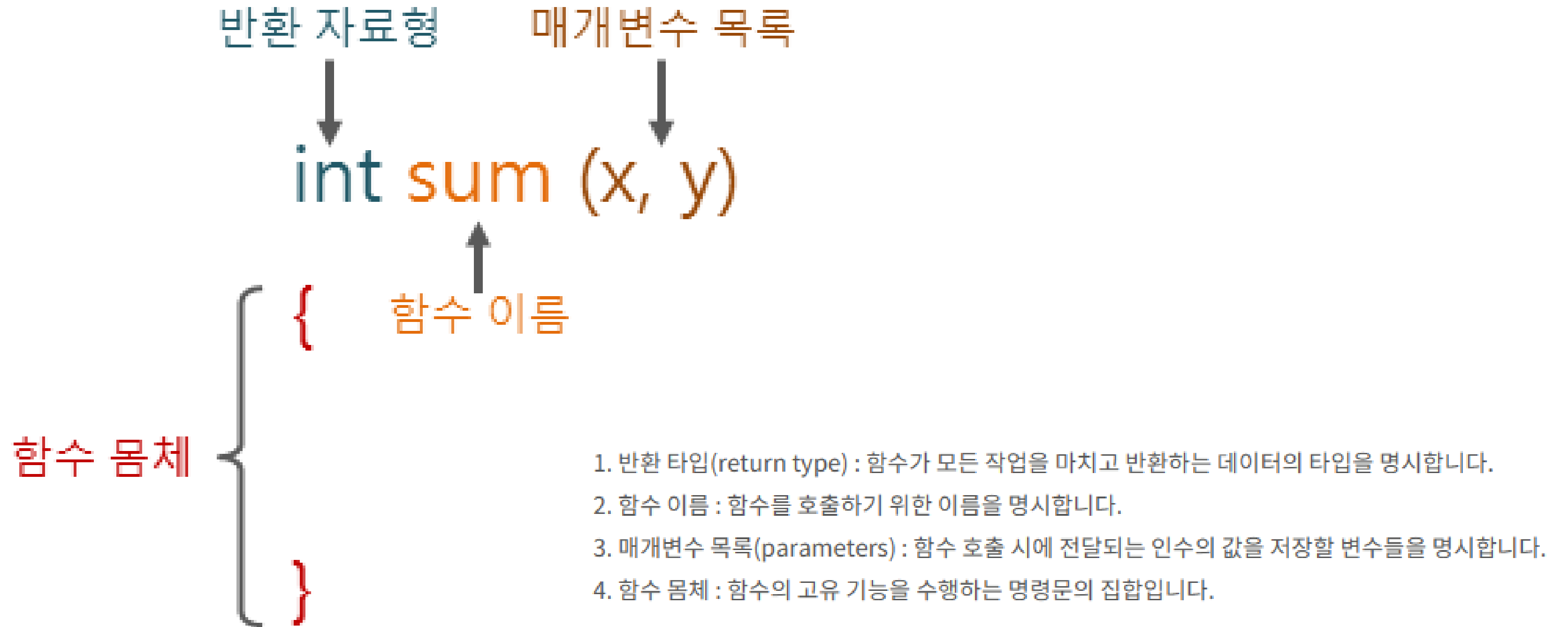
또한, 프로그램을 여러 개의 함수로 나누어 작성하면, 모듈화로 인해 전체적인 코드의 가독성이 좋아집니다.

그리고 프로그램에 문제가 발생하거나 기능의 변경이 필요할 때에도 손쉽게 유지보수를 할 수 있습니다.

함수



함수



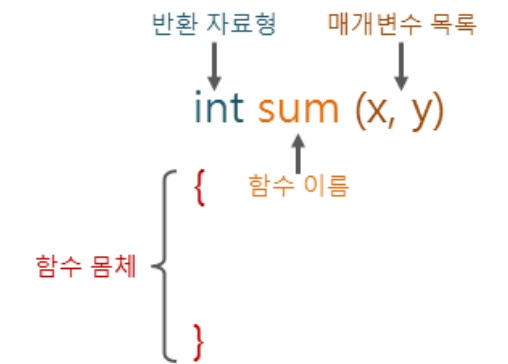
함수

```
int sum(int a, int b){
    return a+b;
}

char return_V(){
    return 'V';
}

string hello(int count_num){
    return to_string(count_num) + "번째 손님 안녕하세요?";
}

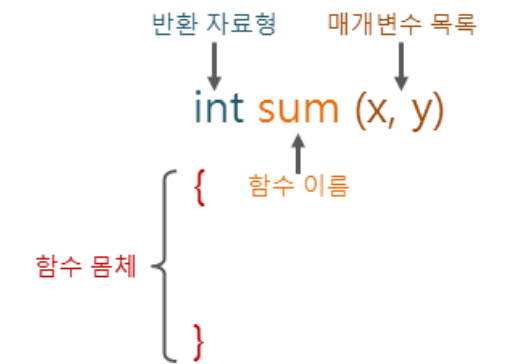
void average(string name, int kor, int mat, int eng) {
    int total = kor + mat + eng;
    double avrage = total / 3.0;
    cout << name << " : " << avrage << " grade";
}
```



1. 반환 타입(return type) : 함수가 모든 작업을 마치고 반환하는 데이터의 타입을 명시합니다.

함수

```
cout << sum(1,3);  
cout << return_V0;  
cout << hello(3);  
average("A",1,1,1);
```



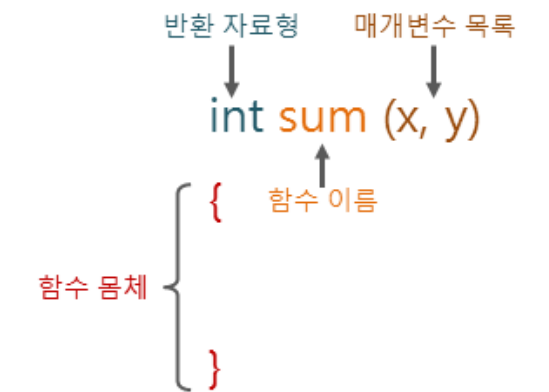
2. 함수 이름 : 함수를 호출하기 위한 이름을 명시합니다.

함수

```
int sum(int a, int b){  
    return a+b;  
}
```

```
char return_V(){  
    return 'V';  
}
```

```
string hello(int count_num){  
    return to_string(count_num) + "번째 손님 안녕하세요?";  
}
```



3. 매개변수 목록(parameters) : 함수 호출 시에 전달되는 인수의 값을 저장할 변수들을 명시합니다.

함수

```
void average(string name, int kor, int mat, int eng) {  
    int total = kor + mat + eng;  
    double avrage = total / 3.0;  
    cout << name << " : " << avrage << " grade";  
}
```

함수

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

int bigNum(int num01, int num02) {
    if (num01 >= num02) {
        return num01;
    } else {
        return num02;
    }
}

int main() {
    int a, b;
    cin >> a >> b;
    cout << "result : " << bigNum(a, b);

    return 0;
}
```

함수

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

int main() {
    int a, b;
    cin >> a >> b;
    cout << "result : " << bigNum(a, b);

    return 0;
}

int bigNum(int num01, int num02) {
    if (num01 >= num02) {
        return num01;
    } else {
        return num02;
    }
}
```

```
:
C:\Users\JH\Documents\K_SukH\algorithm\algo_practice\tndjq.cpp:8:39: error: 'bigNum' was
not declared in this scope
    cout << "result : " << bigNum(a, b);
                              ^
```

함수

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

int bigNum(int num01, int num02);

int main() {
    int a, b;
    cin >> a >> b;
    cout << "result : " << bigNum(a, b);

    return 0;
}

int bigNum(int num01, int num02) {
    if (num01 >= num02) {
        return num01;
    } else {
        return num02;
    }
}
```


함수

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

int what_number(int c) { cout << number * c; }
int main() {
    int number = 3;
    what_number(50);

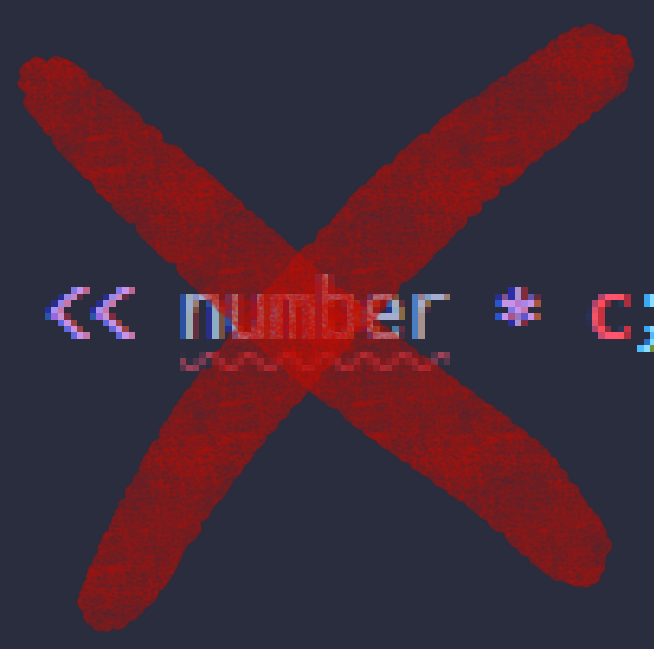
    return 0;
}
```

함수

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

int what_number(int c) { cout << number * c; }
int main() {
    int number = 3;
    what_number(50);

    return 0;
}
```



재귀

옛날옛날 내가 고등학교를 다니던 시절에
선생님께 질문을 했습니다.
"선생님 재귀 함수가 무엇인가요?"
그러자 선생님이 말했습니다.

재귀



재귀



Recursion : 자기 자신을 호출 하는 함수 = 재귀 함수

재귀

```
#include <iostream>

using namespace std;

int recursion_example(int num) {
    if (num == 0)
        return 0;
    else
        return num + recursion_example(num - 1);
}

int main() {
    int answer = recursion_example(5);
}
```

재귀

모든 재귀 호출은 반복문으로 변경 가능하며 그 역으로도 성립한다.
모든 반복문은 재귀 호출로도 변경 가능하다.

왜 재귀함수를 써야 할까?

재귀

복잡한 알고리즘을 사람이 보기에 단순하고 알기 쉽게 표현 가능하기 때문에

단점 : 함수 호출에 따른 오버헤드가 있음

재귀

재귀 함수를 짜는데 꼭 필요한 것

1. 종료 case가 있어야 함 (끝이 있어야 함)
2. 모든 case는 종료 case를 향해야 함
(매개 변수를 잘 활용해야 함)

재귀 예시

팩토리얼

$$5! = 5 \times 4 \times 3 \times 2 \times 1$$

재귀 예시

팩토리얼

$$5! = 5 \times 4 \times 3 \times 2 \times 1$$

종료 조건

1. 1이면 재귀를 종료
2. 모든 숫자는 1을 향하게

```
if (n == 1)  
    return 1;
```

```
else  
    return n * factorial(n - 1);
```

재귀 예시

팩토리얼

$$5! = 5 \times 4 \times 3 \times 2 \times 1$$

종료 조건

1. 1이면 재귀를 종료
2. 모든 숫자는 1을 향하게

```
#include <iostream>

using namespace std;

int factorial(int n) {
    if (n == 1)
        return 1;
    else
        return n * factorial(n - 1);
}

int main() {

    cout << factorial(5);

}
```

오늘의 목표 피보나치 함수 구현

$$F_0 = 0, F_1 = 1, F_{n+2} = F_{n+1} + F_n$$

(0), 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987

종료 조건

문제

기초

10872 팩토리얼

2747 피보나치 수

17478 재귀함수가 뭔가요?

고수

원하는 데이터 재귀로 찾기 - 저한테 검사받기

data[0] ~ data[n-1] 사이에서 target이 있는지 검색한다.
존재하면 target과 일치하는 원소의 인덱스를 리턴, 없으면
-1 을 리턴.

```
int search(int data[], int n, int target)
```

9613 - GCD 합(최대공약수들의 합 구하기)

유클리드 호제법 모르면 검색

11729 - 하노이의 탑 이동 순서

1074 - Z

2448 - 별 찍기 11

재귀

이것들이 재귀함수입니다.
라고 선생님이 말하셨습니다.