

STL과 함수 인자

```
01 struct pt{
02     int x,y;
03 };
04 void func(pt a){
05     a.x = 10;
06 }
07 int main(void) {
08     pt tmp = {0,0};
09     func(tmp);
10     cout << tmp.x;
11 }
```

tmp를
호출한 것이
들어감

0

Swap

```
void swap2(int* a, int* b){
    int tmp = *a;
    *a = *b;
    *b = tmp;
}
```

C++ 참조자(Reference)

int reference

```
void swap3(int& a, int& b){
    int tmp = a;
    a = b;
    b = tmp;
}
```

C++에서 제공되는 라이브러리

STL(Standard Template Library) - vector

```
vector<int> v(100);
v[20] = 10;
v[60] = -4;
```

type이 int이고
0으로 초기화된
100칸짜리 가변배열 v

인덱스 접근
값 변경

Photo

```
01 void func1(vector<int> v) {
02     v[10] = 7;
03 }
04 int main(void) {
05     vector<int> v(100);
06     func1(v);
07     cout << v[10];
08 }
```

복사본

0

```
01 bool cmp1(vector<int> v1, vector<int> v2, int idx) {
02     return v1[idx] > v2[idx];
03 }
```

v1, v2의
크기가 N

복사해서 들어가기 때문에 $O(N)$

```
01 bool cmp2(vector<int>& v1, vector<int>& v2, int idx) {
02     return v1[idx] > v2[idx];
03 }
```

reference로 v1, v2의
주소만 들어가기 때문에 $O(1)$

표준 입출력

```
01 int main(void) {
02     string s = "baaaaaaarkingdog";
03     printf("s is %s\n", s);
04 }
05 /**result**
06 s is ?
07 *****/
```

printf로 string 출력 불가

```
01 int main(void) {
02     char a[10];
03     printf("input : ");
04     scanf("%s", a);
05     string s(a); // 혹은 string s = a;
06     printf("a is %s\n", a); // 기본적인 방식
07     printf("s is %s\n", s.c_str());
08 }
09 /**result**
10 input : test
11 a is test
12 s is test
13 *****/
```

string을
printf로 출력

scanf, cin 둘 다 공백이 나오면 입력이 끝 해결책

```

01 // 1. scanf의 옵션
02 char a1[10];
03 scanf("%[^\\n]", a1);
04
05 // 2. gets 함수(보안상의 이유로 C++14 이상에서는 제거됨)
06 char a2[10];
07 gets(a2);
08 puts(a2);
09
10 // 3. getline 함수 (C++ string)
11 string s;
12 getline(cin, s);
13 cout << s;

```

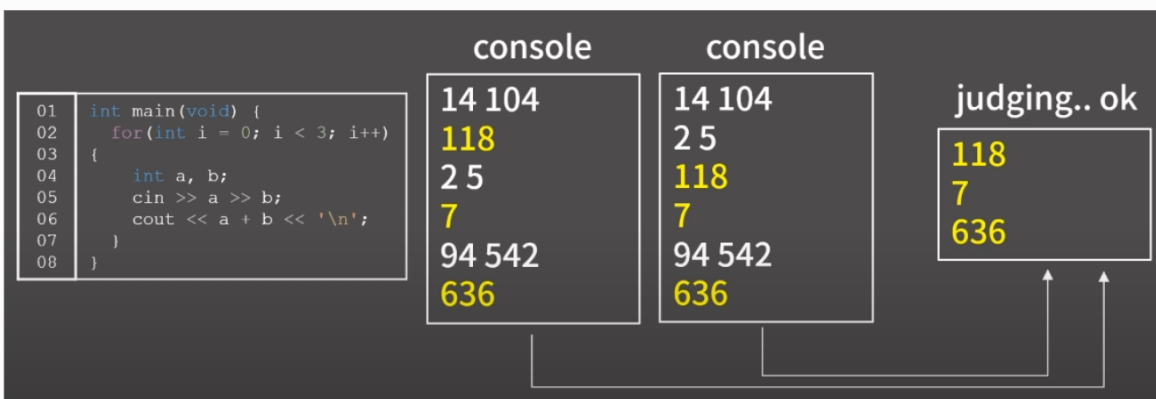
ios::sync_with_stdio(0)

(stream과 C++ stream이 있는데

보통 둘 다 사용할 수 있게 동기화하고 있음.

cout만 사용할 것이라면 굳이 동기화할 필요가
없기 때문에 동기화를 끄음.
→ * cout과 printf를 섞어쓰면 안 됨

cin.tie(0)



cin 명령을 수행하기 전에 cout 버퍼를 비워준다

→ 입출력 순서가 꼬이지 않음 (입력과 출력은 버퍼를 통해 이루어짐)

하지만 online judge 사이트에서는 어차피 출력만 보기 때문에

버퍼를 비울 필요가 없음

