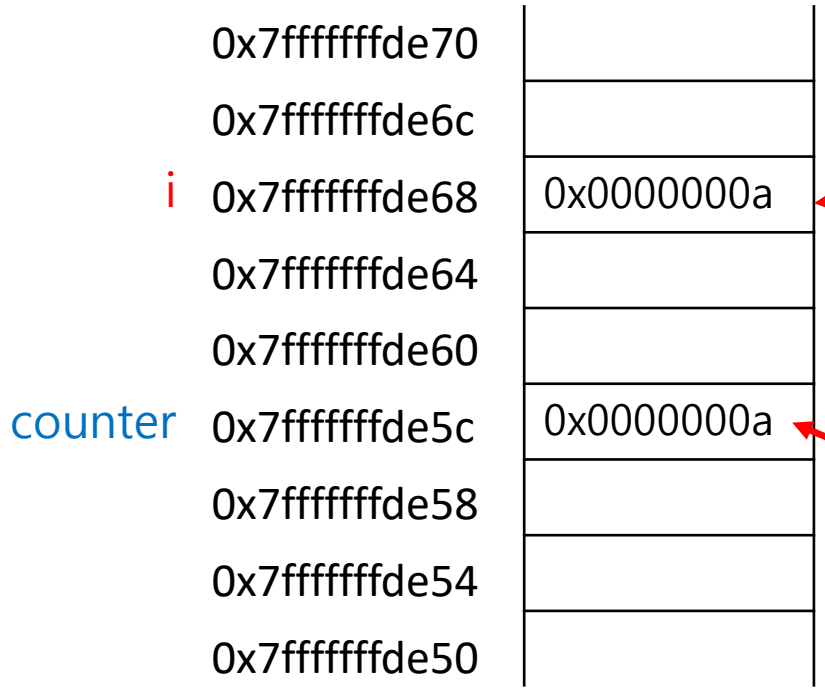
함수 인자의 전달
call by value

2023

국민대학교 소프트웨어학부

C 언어에서 함수의 인자 전달 방법은 call by value



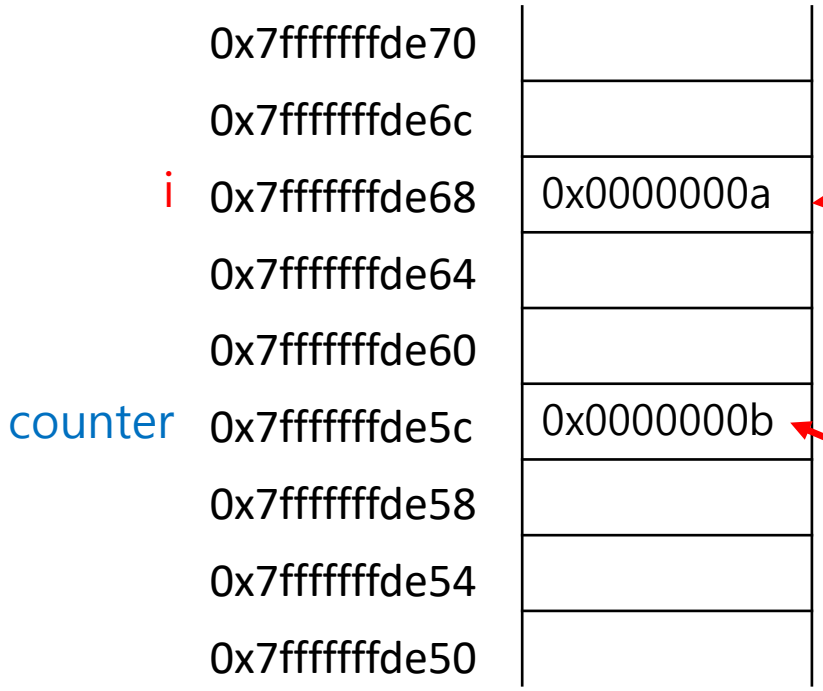

```
#include <iostream>
using namespace std;
int inc(int counter);
int main()
{
    int i;
    i = 10;
    cout << "함수 호출전 i=" << i << endl;
    inc(i);
    cout << "함수 호출후 i=" << i << endl;
    return 0;
}
int inc(int counter)
{
    counter++;
    return counter;
}
```

값에 의한 호출
(call by value)

매개 변수도 일종의 지역 변수임

```
함수 호출전 i=10
```

C 언어에서 함수의 인자 전달 방법은 call by value



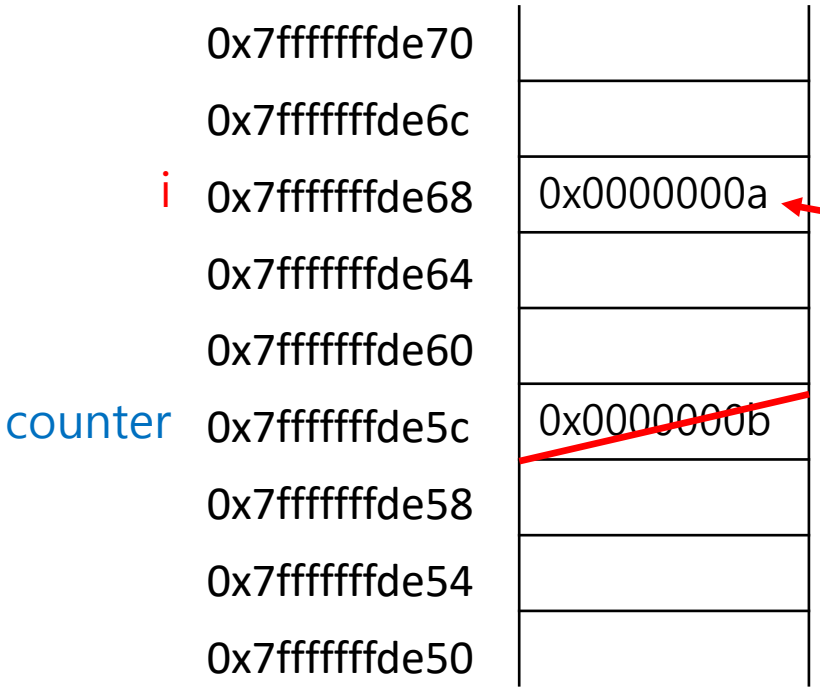

```
#include <iostream>
using namespace std;
int inc(int counter);
int main()
{
    int i;
    i = 10;
    cout << "함수 호출전 i=" << i << endl;
    inc(i);
    cout << "함수 호출후 i=" << i << endl;
    return 0;
}
int inc(int counter)
{
    counter++;
    return counter;
}
```

값에 의한 호출
(call by value)

매개 변수도 일종의 지역 변수임

```
함수 호출전 i=10
```

C 언어에서 함수의 인자 전달 방법은 call by value



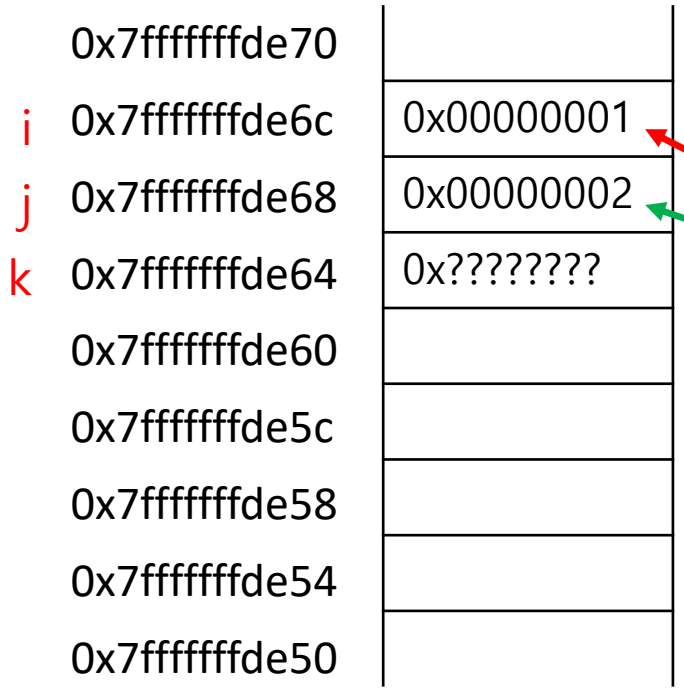

```
#include <iostream>
using namespace std;
int inc(int counter);
int main()
{
	int i;
	i = 10;
	cout << "함수 호출전 i=" << i << endl;
	inc(i);
	cout << "함수 호출후 i=" << i << endl;
	return 0;
}
int inc(int counter)
{
	counter++;
	return counter;
}
```

값에 의한 호출
(call by value)

매개 변수도 일종의 지역 변수임

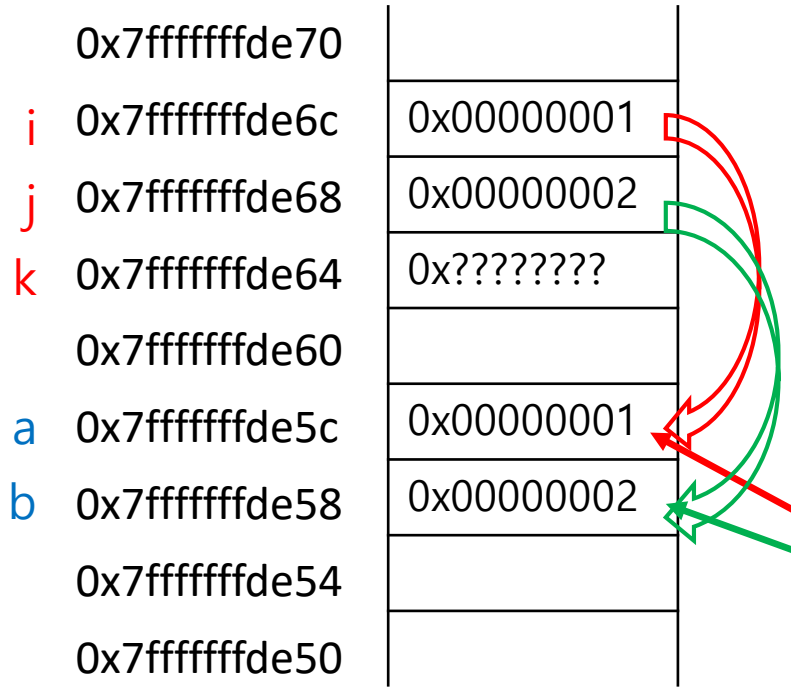
```
함수 호출전 i=10
함수 호출후 i=10
```

C 언어에서 함수의 인자 전달 방법은 call by value



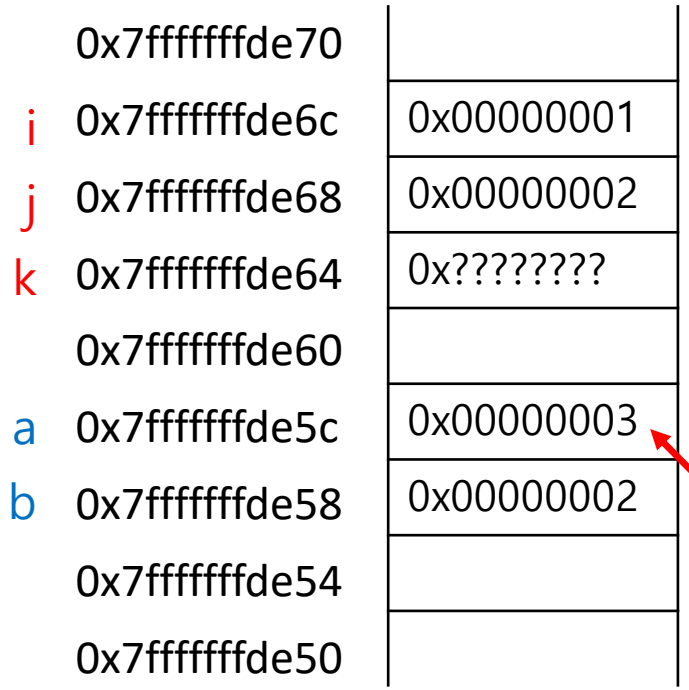

```
1  #include <iostream>
2  using namespace std;
3
4  int f1(int a, int b=0);
5
6  int main(){
7    int   i = 1, j = 2, k;
8
9    k = f1(i, j);
10
11   cout << "k = " << k << endl;
12   return 0;
13 }
14 int f1(int a, int b){
15   a = a+b;
16   return a;
17 }
```

C 언어에서 함수의 인자 전달 방법은 call by value



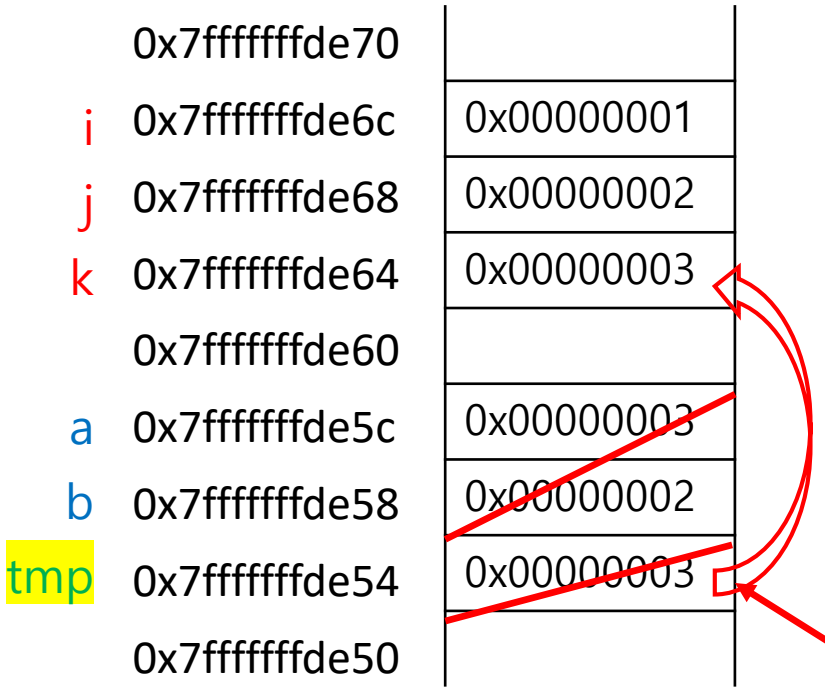

```
1  #include <iostream>
2  using namespace std;
3
4  int f1(int a, int b=0);
5
6  int main(){
7    int   i = 1, j = 2, k;
8
9    k = f1(i, j);
10
11   cout << "k = " << k << endl;
12   return 0;
13 }
14 int f1(int a, int b){
15   a = a+b;
16   return a;
17 }
```

C 언어에서 함수의 인자 전달 방법은 call by value



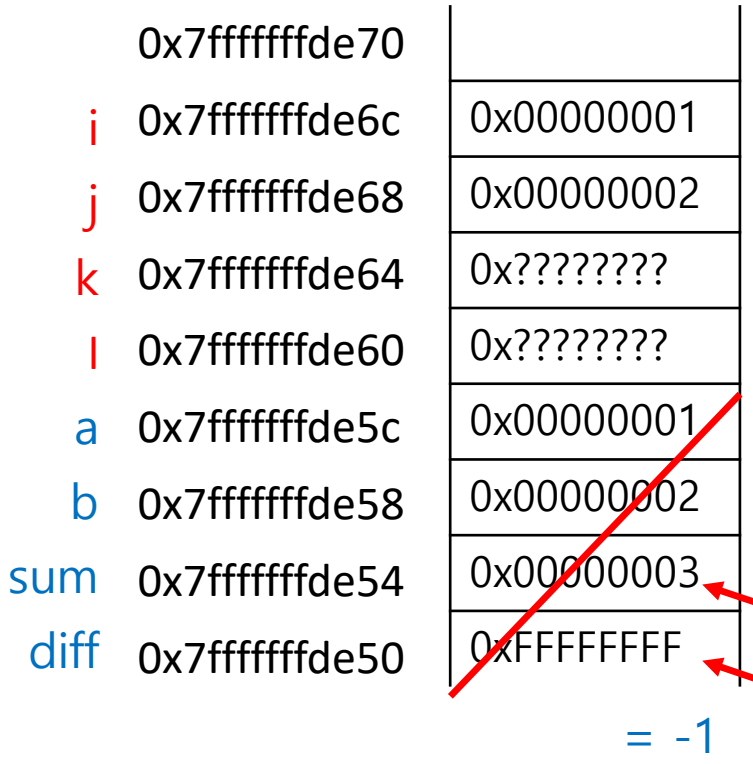

```
1  #include <iostream>
2  using namespace std;
3
4  int f1(int a, int b=0);
5
6  int main(){
7    int   i = 1, j = 2, k;
8
9    k = f1(i, j);
10
11   cout << "k = " << k << endl;
12   return 0;
13 }
14 int f1(int a, int b){
15   a = a+b;
16   return a;
17 }
```

C 언어에서 함수의 인자 전달 방법은 call by value




```
1  #include <iostream>
2  using namespace std;
3
4  int f1(int a, int b=0);
5
6  int main(){
7    int   i = 1, j = 2, k;
8
9    k = f1(i, j);
10
11   cout << "k = " << k << endl;
12   return 0;
13 }
14 int f1(int a, int b){
15   a = a+b;
16   return a;
17 }
```

return 값이 두 개 이상이면?



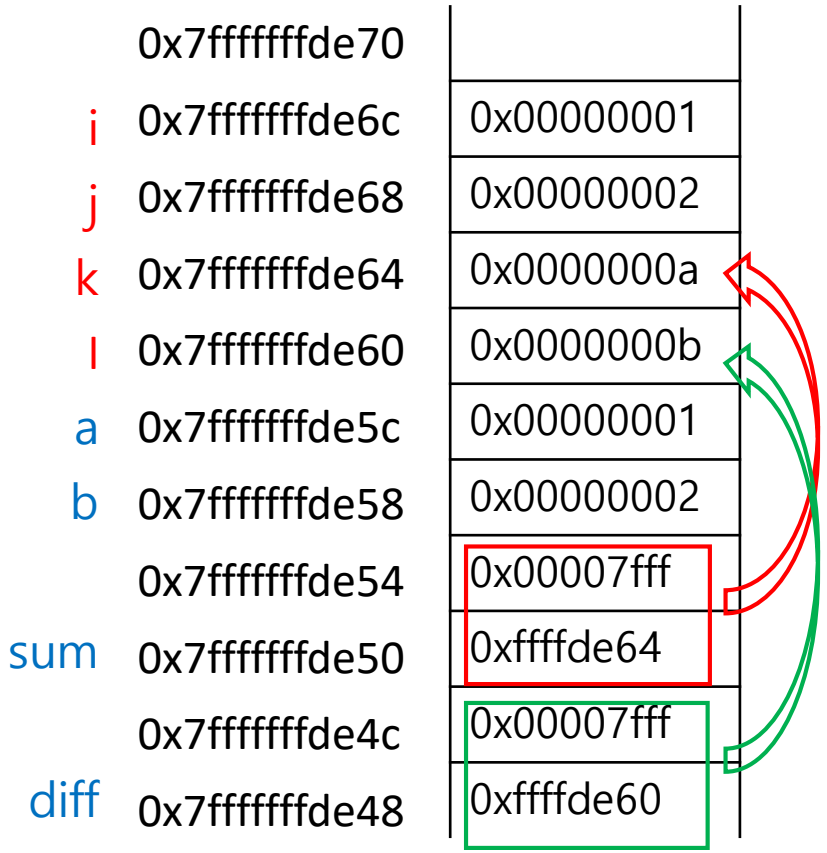

```
1  #include <iostream>
2  using namespace std;
3
4  void f2(int a, int b, int sum, int diff);
5
6  int main(){
7    int   i = 1, j = 2, k, l;
8
9    f2(i, j, k, l);
10
11   cout << "k = " << k << endl;
12   cout << "l = " << l << endl;
13   return 0;
14 }
15 void f2(int a, int b, int sum, int diff){
16   sum = a+b;
17   diff = a-b;
18 }
```

return 값이 두 개 이상이면?

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

int f2(int a, int b=0, int *sum, int *diff);

int main(){
  int   i = 1, j = 2, k= 10, l=11;

  f2(i, j, &k, &l);

  cout << "k = " << k << endl;
  cout << "l = " << l << endl;
  return 0;
}
int f2(int a, int b, int *sum, int *diff){
  *sum = a+b;
  *diff = a-b;
  return *sum;
}
```

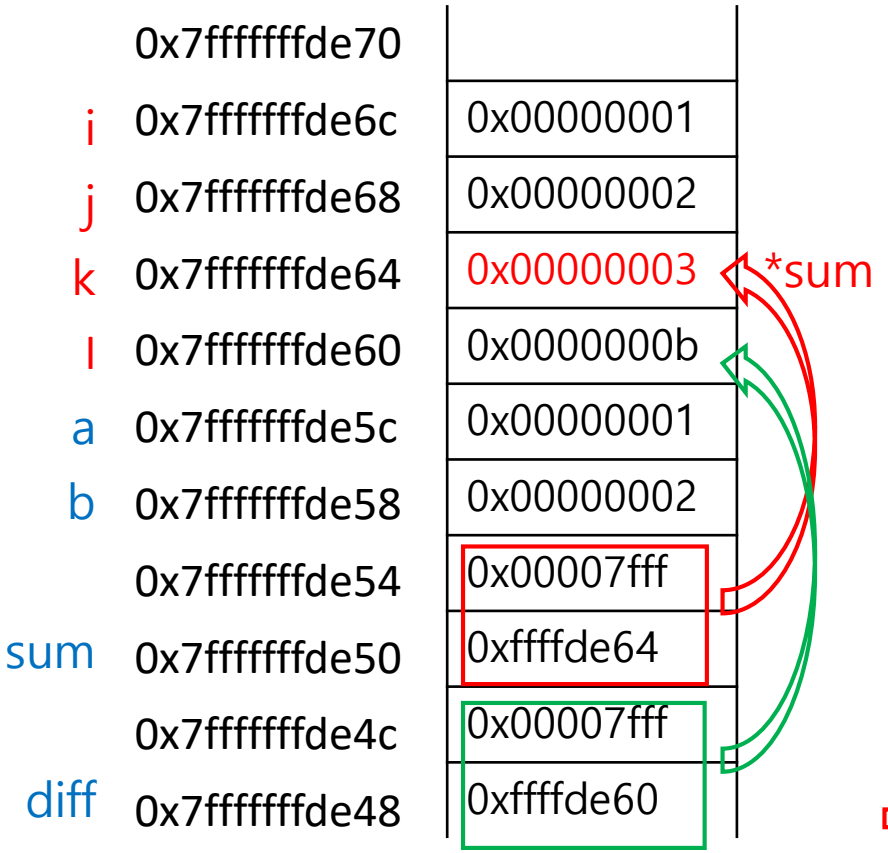
```
ejim@ejim-VirtualBox:~/C2020$ g++ -g -o f2 f2.cpp
f2.cpp:4:5: error: default argument missing for parameter 3 of ‘int f2(int, int, int*, int*)’
 int f2(int a, int b=0, int *sum, int *diff);
     ^~
f2.cpp:4:5: error: default argument missing for parameter 4 of ‘int f2(int, int, int*, int*)’
```

함수의 인자로 pointer type 을 활용하여 call by reference 같은 효과



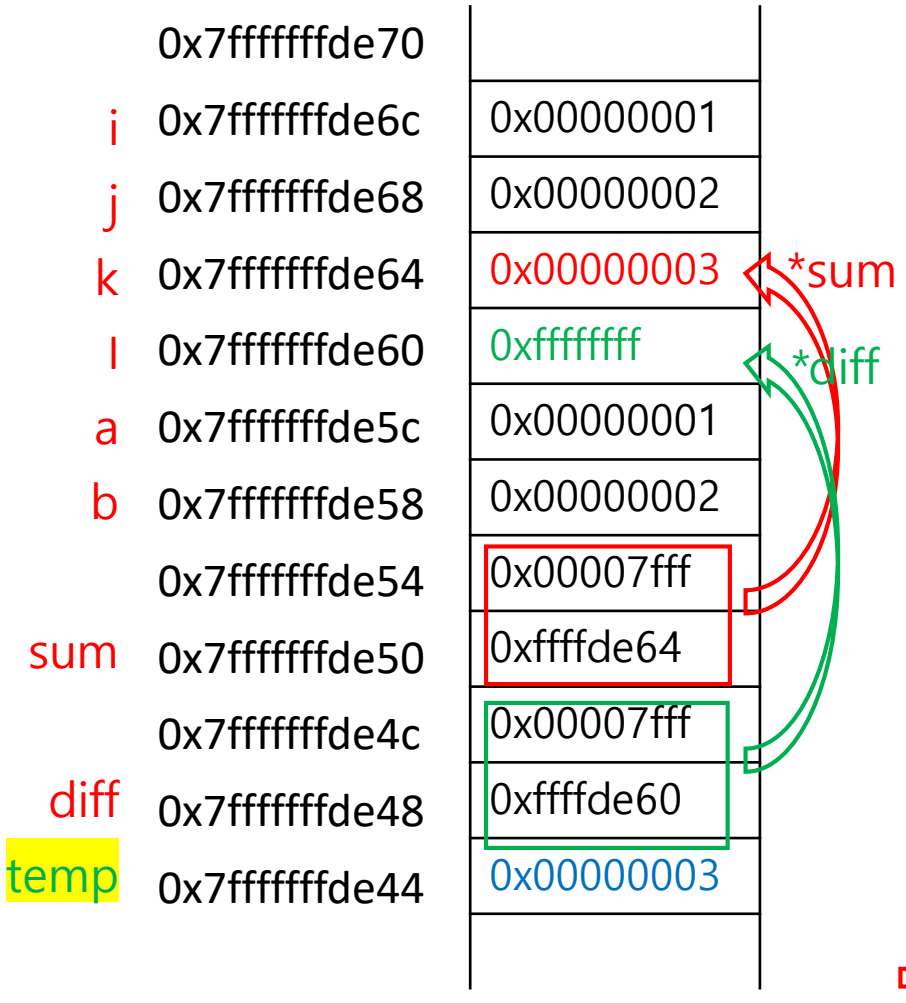

```
1  #include <iostream>
2  using namespace std;
3
4  int f2(int a, int b=0, int *sum=nullptr, int *diff=nullptr);
5
6  int main(){
7    int   i = 1, j = 2, k= 10, l=11;
8
9    f2(i, j, &k, &l);
10
11   cout << "k = " << k << endl;
12   cout << "l = " << l << endl;
13   return 0;
14 }
15 int f2(int a, int b, int *sum, int *diff){
16   *sum = a+b;
17   *diff = a-b;
18   return *sum;
19 }
```

함수의 인자로 pointer type 을 활용하여 call by reference 같은 효과



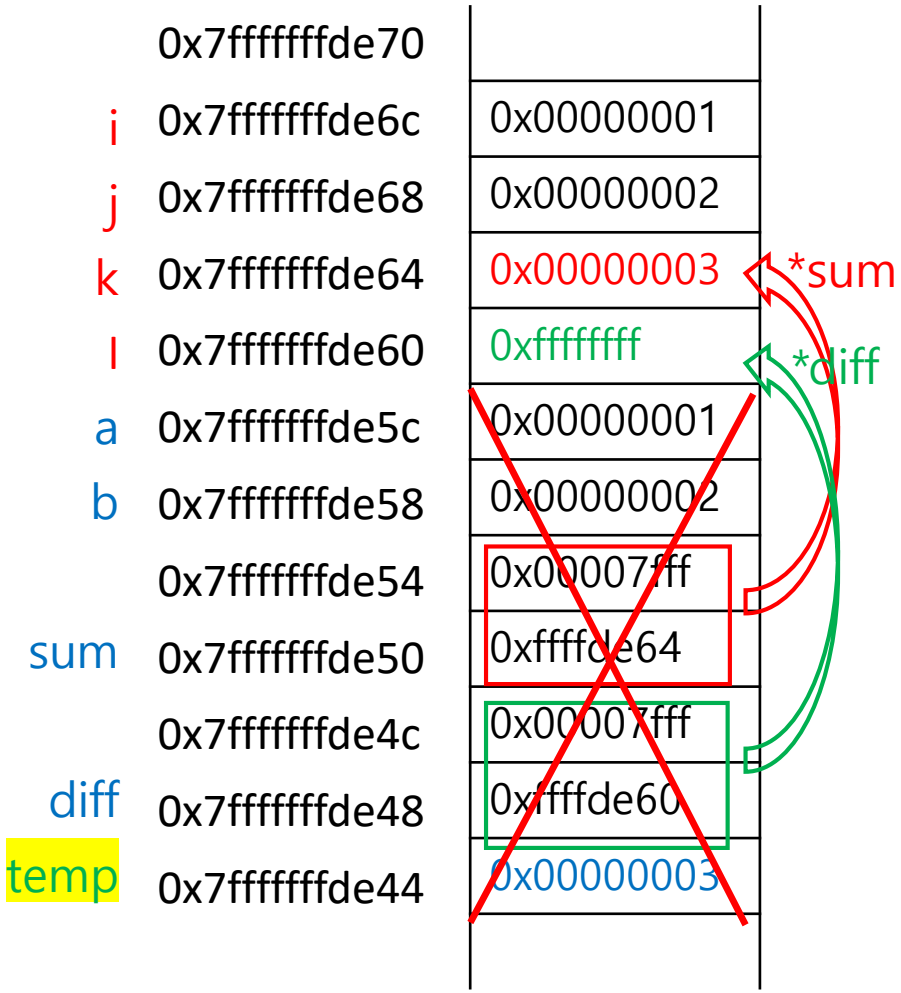

```
1  #include <iostream>
2  using namespace std;
3
4  int f2(int a, int b=0, int *sum=nullptr, int *diff=nullptr);
5
6  int main(){
7    int   i = 1, j = 2, k= 10, l=11;
8
9    f2(i, j, &k, &l);
10
11   cout << "k = " << k << endl;
12   cout << "l = " << l << endl;
13   return 0;
14 }
15 int f2(int a, int b, int *sum, int *diff){
16   *sum = a+b;
17   *diff = a-b;
18   return *sum;
19 }
```

함수의 인자로 pointer type 을 활용하여 call by reference 같은 효과




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

int f2(int a, int b=0, int *sum=nullptr, int *diff=nullptr);

int main(){
  int   i = 1, j = 2, k= 10, l=11;

  f2(i, j, &k, &l);

  cout << "k = " << k << endl;
  cout << "l = " << l << endl;
  return 0;
}
int f2(int a, int b, int *sum, int *diff){
  *sum = a+b;
  *diff = a-b;
  return *sum;
}
```

함수의 인자로 pointer type 을 활용하여 call by reference 같은 효과




```
1  #include <iostream>
2  using namespace std;
3
4  int f2(int a, int b=0, int *sum=nullptr, int *diff=nullptr);
5
6  int main(){
7    int   i = 1, j = 2, k= 10, l=11;
8
9    f2(i, j, &k, &l);
10
11   cout << "k = " << k << endl;
12   cout << "l = " << l << endl;
13   return 0;
14 }
15 int f2(int a, int b, int *sum, int *diff){
16   *sum = a+b;
17   *diff = a-b;
18   return *sum;
19 }
```

nicer handling of pointer arguments

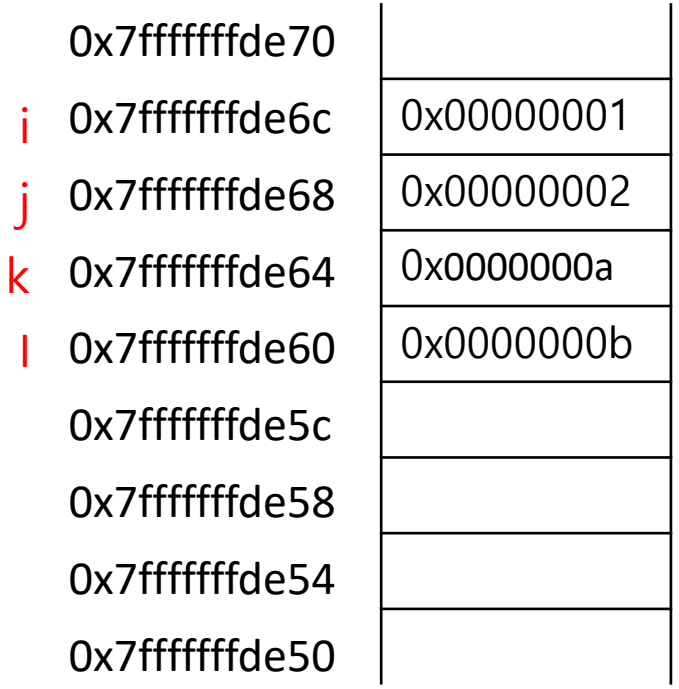
sum, diff 가 null pointer 일 때

```
1  #include <iostream>
2  using namespace std;
3
4  int f2(int a, int b=0, int *sum=nullptr, int *diff=nullptr);
5
6  int main(){
7    int   i = 1, j = 2, k= 10, l=11;
8
9    f2(i, j, &k, &l);
10
11   cout << "k = " << k << endl;
12   cout << "l = " << l << endl;
13   return 0;
14 }
15 int f2(int a, int b, int *sum, int *diff){
16   *sum = a+b;
17   *diff = a-b;
18   return *sum;
19 }
```

```
ejim@ejim-VirtualBox:~/C2020$ ./f3
Error : sum is a null pointer.
```

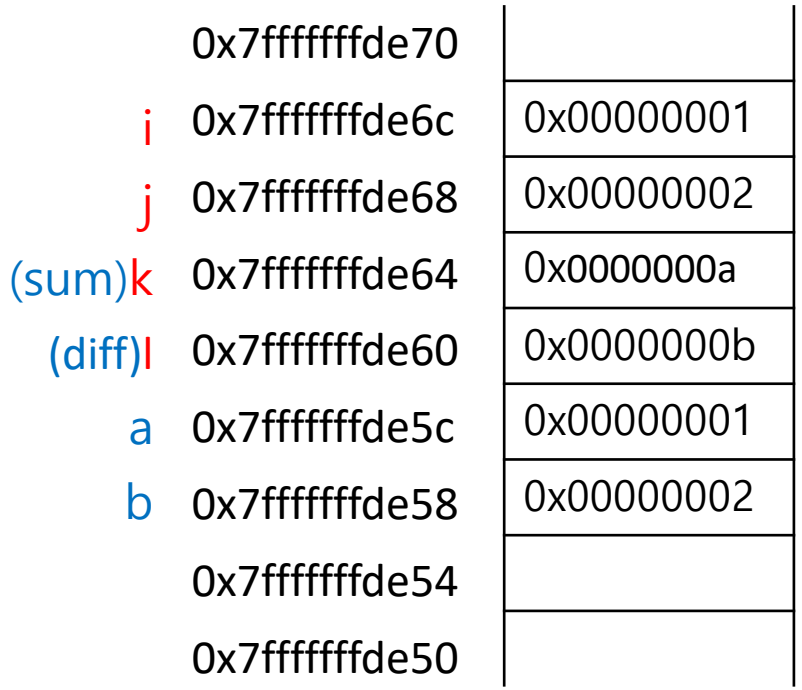
```cpp
#include <iostream>
#include <climits>
using namespace std;

int f2(int a, int b=0, int *sum=nullptr, int *diff=nullptr);
int main(){
  int   i=1, j=2, k=10, l=11, r=12;
  r = f2(i, j);
  if (r==INT_MIN) return 1;

  cout << "k = " << k << endl;
  cout << "l = " << l << endl;
  return 0;
}
int f2(int a, int b, int *sum, int *diff){
  if (sum==nullptr){
     cout<< "Error : sum is a null pointer.\n";
     return INT_MIN;
  }
  if (!diff){ // same as if(diff==nullptr)
     cout<< "Error : diff is a null pointer.\n";
     return INT_MIN;
  }
  *sum = a+b;
  *diff = a-b;
  return *sum;
}
```

reference 선언 : 선언문의 &




```
1  #include <iostream>
2  using namespace std;
3
4  int f3(int a, int b, int &sum, int &diff);
5
6  int main(){
7    int   i=1, j=2, k=10, l=11;
8
9    f3(i, j, k, l);
10
11   cout << "k = " << k << endl;
12   cout << "l = " << l << endl;
13   return 0;
14 }
15 int f3(int a, int b, int &sum, int &diff){
16   sum = a+b;
17   diff = a-b;
18   return sum;
```

reference 선언 : 선언문의 &



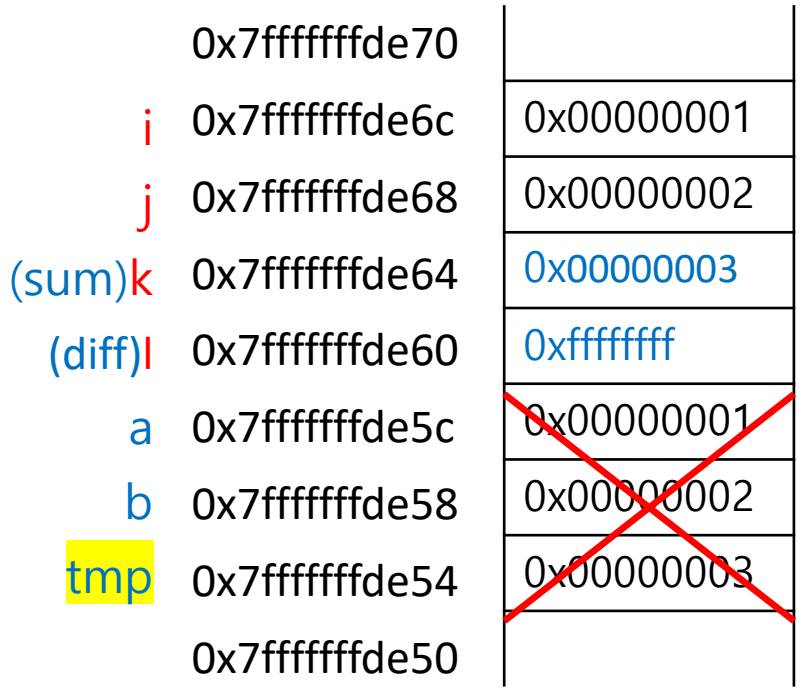

```
1  #include <iostream>
2  using namespace std;
3
4  int f3(int a, int b, int &sum, int &diff);
5
6  int main(){
7    int   i=1, j=2, k=10, l=11;
8
9    f3(i, j, k, l);
10
11   cout << "k = " << k << endl;
12   cout << "l = " << l << endl;
13   return 0;
14 }
15 int f3(int a, int b, int &sum, int &diff){
16   sum = a+b;
17   diff = a-b;
18   return sum;
```

reference 선언 : 선언문의 &

	0x7fffffffde70	
i	0x7fffffffde6c	0x00000001
j	0x7fffffffde68	0x00000002
(sum)k	0x7fffffffde64	0x00000003
(diff)l	0x7fffffffde60	0xffffffff
a	0x7fffffffde5c	0x00000001
b	0x7fffffffde58	0x00000002
tmp	0x7fffffffde54	0x00000003
	0x7fffffffde50	

```
1  #include <iostream>
2  using namespace std;
3
4  int f3(int a, int b, int &sum, int &diff);
5
6  int main(){
7    int   i=1, j=2, k=10, l=11;
8
9    f3(i, j, k, l);
10
11   cout << "k = " << k << endl;
12   cout << "l = " << l << endl;
13   return 0;
14 }
15 int f3(int a, int b, int &sum, int &diff){
16   sum = a+b;
17   diff = a-b;
18   return sum;
```

reference 선언 : 선언문의 &




```
1  #include <iostream>
2  using namespace std;
3
4  int f3(int a, int b, int &sum, int &diff);
5
6  int main(){
7    int   i=1, j=2, k=10, l=11;
8
9    f3(i, j, k, l);
10
11   cout << "k = " << k << endl;
12   cout << "l = " << l << endl;
13   return 0;
14 }
15 int f3(int a, int b, int &sum, int &diff){
16   sum = a+b;
17   diff = a-b;
18   return sum;
```

Reference 의 사용 (함수 인자로만 사용하자 ㅜㅜ)

```
int  i = 10;
int &r = i;  // reference 는 선언문에서만 초기화가 가능하다.
             // int &r;  int &r=10;  모두 compile error!

r = i;  // error 는 아니지만 의미가 없는 문장임
r = 20;  // i 의 값이 20 이 됨

cout << i << endl;  //  20 이 출력
```

변수의 aliases : alias를 통해 변수 값을 바꿀 수 있다.

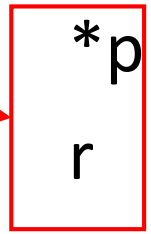
```
int i;
int *p;
p = &i;
int &r = i;
```

실행문의 &는 주소 생성 연산자

선언문의 &는 reference 선언

```
i  = 1;
*p = 2;
r  = 3;
```

모두 변수 i 의 aliases



reference 인자의 초기화

```
1  #include <iostream>
2  using namespace std;
3
4  int f3(int a, int b, int &sum, int &diff);
5
6  int main(){
7    int   i=1, j=2, k=10, l=11;
8
9    f3(i, j, k, l);
10
11   cout << "k = " << k << endl;
12   cout << "l = " << l << endl;
13   return 0;
14 }
15 int f3(int a, int b, int &sum, int &diff){
16   sum = a+b;
17   diff = a-b;
18   return sum;
```

함수 호출 시 parameter initialization

int a = i;

int b = j;

int &sum = k;

int &diff = l;

pointer 인자의 초기화

```
1  #include <iostream>
2  using namespace std;
3
4  int f2(int a, int b=0, int *sum=nullptr, int *diff=nullptr)
5
6  int main(){
7    int   i = 1, j = 2, k= 10, l=11;
8
9    f2(i, j, &k, &l);
10
11   cout << "k = " << k << endl;
12   cout << "l = " << l << endl;
13   return 0;
14 }
15 int f2(int a, int b, int *sum, int *diff){
16   *sum = a+b;
17   *diff = a-b;
18   return *sum;
19 }
```

함수 호출 시 parameter initialization

int a = i;

int b = j;

int *sum = &k;

int *diff = &l;