



# 20191107 실습 7주차

박건호

KITRI BoB 8<sup>th</sup>

5422 RTDCS

[devgunho.github.io](https://devgunho.github.io)



# 1. CPointerReview (5)

size와 value를 입력받아 size만큼 value를 담을 수 있는 공간을 동적 할당하는 함수를 구현하려 한다.  
하지만 allocateArray에서 정상적으로 입력값만큼 할당을 했음에도,  
출력이 제대로 되지 않는다.

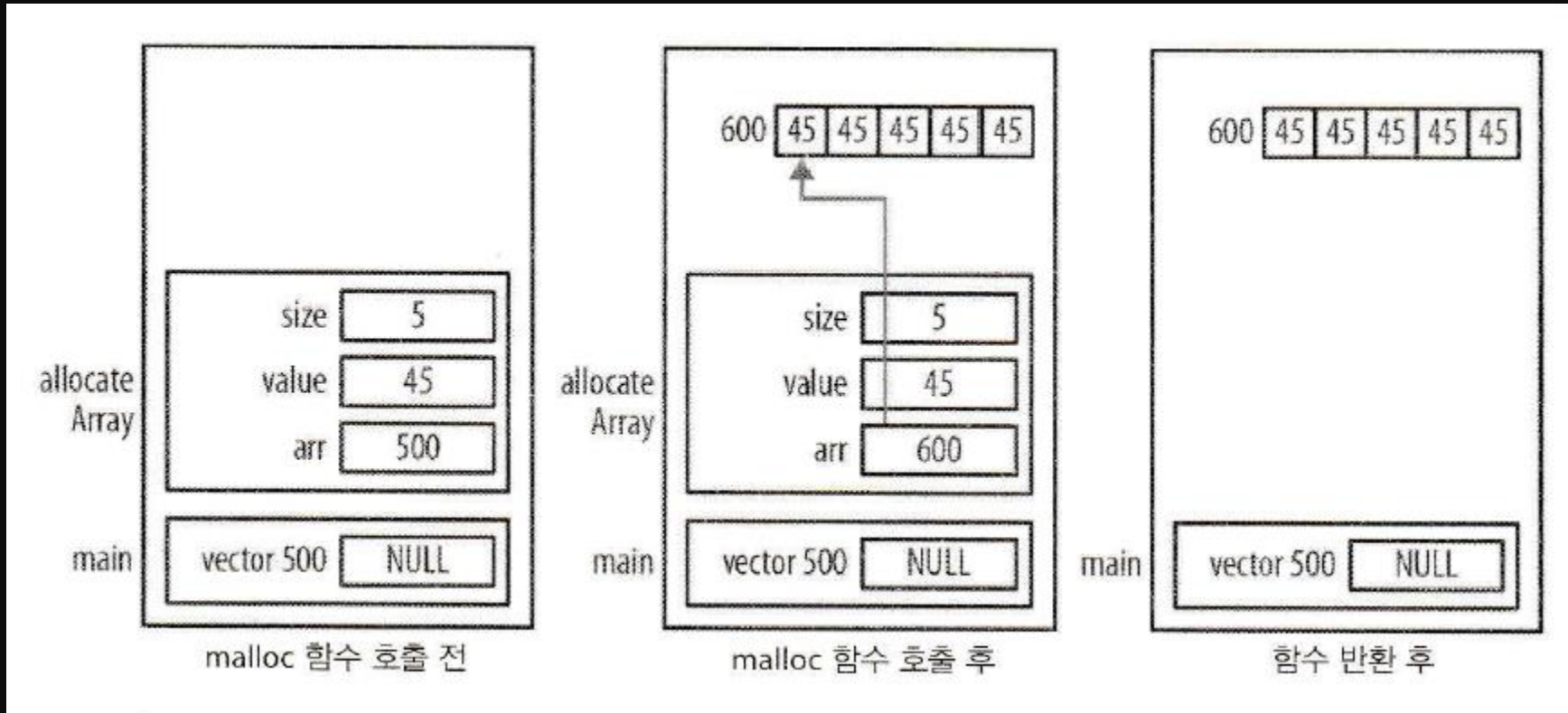
이유가 무엇인지 생각해보고 코드를 수정해서 올바르게 동작하도록 한다.

(문제의 코드는 C++이 아닌 C이다.)

```
Microsoft Visual Studio Debug Console
10 2
2 2 2 2 2 2 2 2 2 2
D:\W_Workspace\19_ComputerProgramming\CP_Le
Press any key to close this window . . .
```

# 1. CPointerReview

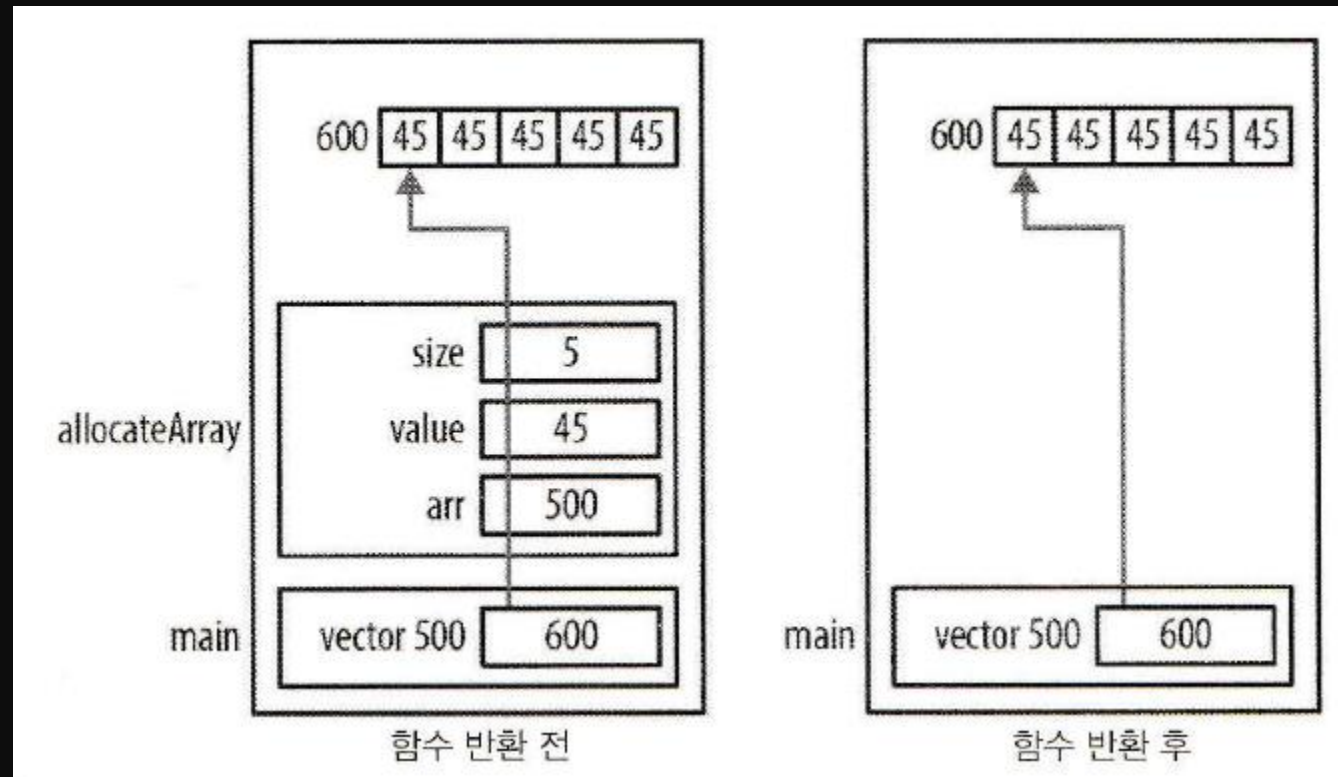
문제의 코드를 그림으로 설명하면 다음과 같다.





# 1. CPointerReview

다음과 같은 상황으로 정상 동작하도록 바꾸어 주어야한다.





## 2. FindNthVal (10)

**STL을 사용하지 않고**, -1 이 입력될 때까지 입력된 값 중 N번째 값을 찾기

-1이 입력될 때까지 하나씩 입력

입력값의 범위는  $0 \leq \text{input} \leq 500$  만약 범위가 아닌 수가 입력된다면 무시하고 계속 진행

-1이 입력된 이후엔, 찾고자 하는 값 N 입력

- Vector 사용 X

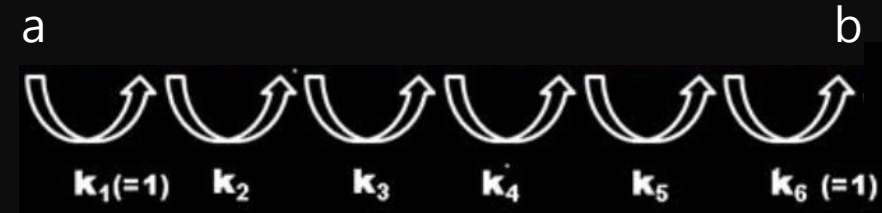
- C로 구현

Hint : 몇 개의 값이 입력될지 모르기 때문에 동적할당 필요

Hint : 반복문 안에서 동적할당과 free를 반복해준다.

```
Microsoft Visual Studio
900
900
900
5
4
3
2
500
9999
-1
5
500
D:\Workspace\19_C
Press any key to c
```

# 3. JumpingMachine



출발지점 a 에서 b까지 ( $a \leq x < y < 2^{31}$ ) 이동하기 위해 JumpingMachine을 타고 이동한다.

이 기계로 움직일 수 있는 거리는 한번에 K만큼 움직일 수 있다고 가정한다.

또한 이전 작동시기에 K만큼 이동하였을 때는 K-1, K, K+1 만큼 이동할 수 있다고 가정한다.

예를 들어 처음 이 기계로 움직이는 경우 -1, 0, 1 만큼 이동할 수 있으나

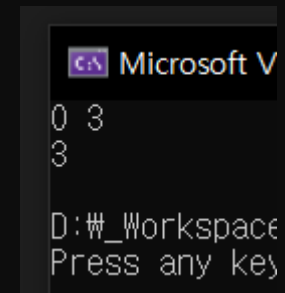
사실상 음수나 0의 거리만큼의 거리 이동은 의미가 없으므로 1만큼 이동이 가능하고,

그 다음에는 0, 1, 2만큼 이동이 가능하다.

여기서 다시 2만큼 이동한다면 다음 시기엔 1, 2, 3 만큼 이동이 가능하다.

a지점에서 b지점까지 최소한의 작동 횟수로 이동하려 한다.

b지점에 도착하기 바로 직전의 이동거리는 반드시 1만큼으로 고정한다.





## 4. PointSwap

코드가 정상 동작하도록 구현하기 (C코드)

Point struct 를 구현하고, (typedef 지정)

struct 내부에는 정수 x, y, z의 좌표를 갖을 수 있도록 구현한다.

```
Point pos1, pos2;    // 두 점 선언
```

```
SwapPoint2Point(&pos1, &pos2);    // 두 좌표 Swap
```

```
SwapXZ(&pos1.xpos, &pos1.zpos);    // 한 좌표의 점x 와 점z Swap
```



# 5. StringDeepCopy

깊은 복사를 하도록 하는 코드가 정상 동작하도록 구현하기

참고 :

```
A(char* x)
{
    cout << "char* Parametrized Constructor called!\n";
    int l = strlen(x), i;
    s = new char[l + 1];
    for (i = 0; i < l; i++)
        s[i] = x[i];
    s[l] = '\0';
}
```

Microsoft Visual Studio Debug Console

```
aaabbbcccddd
string Parametrized Constructor called!
Copy Constructor called!
aaabbbcccddd
```





# 6. Company

회사의 부서(Department)와 부서 내의 구성원(Employee)을 나타내는 클래스를 구현하여  
다음 출력을 만족시키기

## Employee 클래스

private:

- name, ID

public:

- 생성자
- setName
- getName
- setID
- getID

## Department 클래스

private:

- string name;
- Employee\* TeamLeader;

public:

- 생성자
- setTeamLeader
- print



# 객체 포인터 활용 이유

아래 코드와 같이 클래스 안에 다른 클래스 객체를 멤버 변수로 가지고 있는 경우에 TeamLeader가 없는 경우에도 Employee 클래스 전체를 가지고 있다면, 매우 비효율적이라고 할 수 있다.

```
int main()
{
    string str1, str2, str3;
    cin >> str1 >> str2 >> str3;
    Department dp(str1);
    Employee* p = new Employee(str2, str3);
    dp.setTeamLeader(p);
    dp.print();

    p = nullptr;
    dp.setTeamLeader(p);
    dp.print();

    delete p;
    return 0;
}
```

## Microsoft Visual Studio Debug Console

```
인사과 홍길동 ID1234
인사과 팀장은 홍길동
팀장의 ID는 ID1234
인사과 팀장은 현재 없습니다.
```

```
D:\#_Workspace#19_ComputerProgramming#19_CF
h code 0.
Press any key to close this window . . .
```