

소스 구현 설명

문제 정의 :

컴퓨터의 주기억장치를 모델링하는 클래스 Ram을 구현하는 것.

Ram 클래스에는 데이터가 기록될 **메모리 공간(mem[])**과 **크기 정보(size)**를 가지고,

주어진 주소에 데이터를 기록하고(**write**),

주어진 주소로부터 데이터를 읽어 온다(**read**).

<Ram.h>

```
class Ram {
    char mem[100*1024];    //메모리 공간
    int size;              //크기 정보
public:
    Ram();                //기본 생성자
    ~Ram();               //소멸자
    char read(int address); //2. address에 저장되어있는 value를 읽음.
    void write(int address, char value); //1. address에 value를 저장
};
```

main() 함수는 100번지에 20을 저장하고, 101번지에 30을 저장한 후,
100번지와 101번지의 값을 읽고 더하여 102번지에 저장하는 코드이다.

<main.cpp>

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

#include "Ram.h"

int main() {
    Ram ram;
    ram.write(100, 20);
    ram.write(101, 30);
    char res = ram.read(100) + ram.read(101);
    ram.write(102, res);
    cout << "102 번지의 값 = " << (int)ram.read(102) << endl;
}
```

<Ram.cpp>

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

#include "Ram.h"

Ram::Ram(){
    size = 100 * 1024;      //size 100*1024로 초기화
    for (int i = 0; i < size; i++) {    //배열 0으로 초기화
        mem[i] = 0;
    }
}

Ram::~Ram() {    //메모리 제거
    cout << "메모리 제거됨" << endl;
}

char Ram::read(int address) {
    //write에서 mem[주소] = 값 저장했으니, read에서는 그대로 읽기만
    return mem[address];
}

void Ram::write(int address, char value) {
    //mem[주소] = 값 형태로 저장
    mem[address] = value;
}
```

★문제를 해결한 키 아이디어 또는 알고리즘 설명 :

문제를 해결한 키는 배열을 0으로 초기화하는 것과 `char Ram::read`, `void Ram::write` 구현이 있다.

배열 0으로 초기화는 배열의 특징을 활용한 for문으로 초기화를 하였다.
`read`와 `write`는 배열 안에 `mem[주소 값]` 안에 `value`를 읽어온다는 메커니즘으로 해결했다.

★ 문제 해결 방법 문제를 해결하기 위한 아이디어들 :

위에서 설명한 것과 마찬가지로 for문을 통한 배열 초기화와 배열 값에 주소를 넣는다면 서로 다른 공간에 `value` 값을 넣을 수 있다는 아이디어로 문제를 해결하였다.

★ 아이디어 평가 :

for문의 범위를 `size`까지로 해서 배열 공간을 만들었고,
너무 많은 공간을 만들어서 더 효율적인 방법이 있는지 모색하였지만, 문제에 충실이 이행하면 이것밖에 없다고 판단했다.