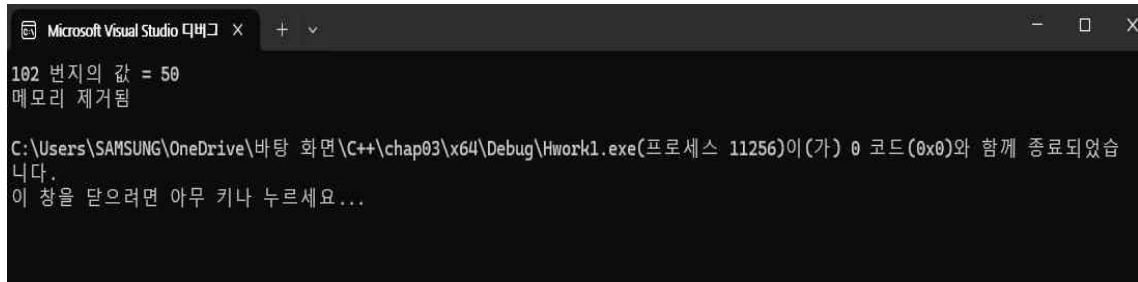


○ 소스 수행 결과 화면



```
Microsoft Visual Studio 디버그
102 번지의 값 = 50
메모리 제거됨
C:\Users\SAMSUNG\OneDrive\바탕 화면\C++\chap03\x64\Debug\Hwork1.exe(프로세스 11256)이(가) 0 코드(0x0)와 함께 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요...
```

○ 소스 구현 설명

문제에 나와있는 main() 함수 코드와 Ram클래스를 통해 생성자 함수 Ram()과 소멸자 함수 ~Ram() 함수를 구현해야 함을 알 수 있다. 또한 read(int address) 메소드와 write(int address, char value) 메소드를 구현해야 한다.

Ram.h, Ram.cpp, main.cpp 총 3개의 파일로 코드를 나누어야 한다.

먼저, main.cpp 파일에는 문제의 main() 함수 코드와 동일하게 코드를 작성하고 저장해준다. main() 함수를 변경하지 않고 문제의 실행 결과와 같게 하기 위함이다.

Ram.h 파일 역시 문제에 제시된 Ram 클래스 코드를 동일하게 작성한다. 다음으로 작성할 Ram.cpp 파일에서 생성자 함수와 메소드 등을 구현하기 위한 틀이 될 것이다.

Ram.cpp 코드 작성에 들어가기 앞서, main.cpp 파일과 Ram.cpp 파일 코드 상단에

```
#include "Ram.h"
```

위와 같은 코드를 작성해주어 Ram 클래스와 메소드를 사용할 수 있게 만든다.

이제 헤더파일을 포함한 Ram.cpp 코드를 작성한다. 다음은 생성자 함수 Ram()을 작성한 것이다.

```
Ram::Ram() {
    size = 100 * 1024;
    mem[size] = {};
}
```

문제에 주석으로 설명하고 있는 size 100*1024 초기화와 새로운 Ram객체가 생성될 때마다 size 크기 만큼의 배열을 비어있는 배열로 초기화하는 코드를 작성했다.

소멸자 함수 ~Ram()에서는 지정할 변수 등 상관없이 생성된 객체가 소멸 될 때 “메모리 제거됨” 문자열을 출력하게 하기 위해 다음과 같이 작성했다.

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

read 메소드는 매개변수로 입력받은 주소의 메모리 바이트를 리턴하기 위해 mem[] 배열에 매개변수를 할당하여 원하는 값을 얻을 수 있도록 하였다.

```
char Ram::read(int address) {  
    return mem[address];  
}
```

write 메소드는 address로 입력받은 주소에 매개변수로 할당된 value 값을 저장하게 해야 한다. 따라서 다음과 같이 작성했다.

```
void Ram::write(int address, char value) {  
    mem[address] = value;  
}
```

Ram.cpp를 모두 작성한 후, main.cpp 파일을 실행하면 앞서 캡처 화면과 같은 결과를 얻을 수 있다.

문제를 해결한 키는 ‘size’에 있다. 문제를 해결해가면서 size를 어떤 용도로 활용해야 할지 의문이었다. 배열을 100*1024 라는 고정크기를 갖게하며 0으로 초기화 해야 한다는 것을 통해 해결할 수 있었다. 배열의 크기 값을 size로 주고, size를 100*1024의 고정크기를 갖도록 변수 설정을 하면 되는 것이었다. 이를 통해, 문제의 실행 결과와 같은 결과화면을 도출할 수 있었다.