What’s wrong with this code?

```
#include <stdlib.h>
#include <sys/mman.h>
#define MAX_MEM 1<<30

static void *base = mmap(NULL, MAX_MEM, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANON, 0, 0);

void *my_buddy_malloc(int size) {
}

void my_free(void *ptr) {
}
```

```
>> gcc -c mymalloc.c
mymalloc.c:5: error: initializer element is not constant

• ‘base’ is allocated in the data section of the object file.
• The initialization value of ‘base’ must be known at compile time to write into object file.
• The return value of mmap is a runtime value.
```
Correct Code

```c
#include <stdlib.h>
#include <sys/mman.h>
#include <assert.h>
#define MAX_MEM 1<<30

static void *base = NULL;
void *my_buddy_malloc(int size) {
    if(base == NULL) {
        base = mmap(NULL, MAX_MEM, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANON, 0, 0);
    }
    assert(base != NULL);
}
void my_free(void *ptr) {
    assert(base != NULL);
}
```

>> gcc -c mymalloc.c
mymalloc.c:5: error: initializer element is not constant

- ‘base’ is only allocated the first time `my_buddy_malloc` is called.
- The subsequent calls start to use the originally allocated block.
- ‘assert’ prints an error message and halts the program if the condition is not true.