Dry

2.1.1. The original code-

```
#include "stdlib.h"
#include "string.h"
#include "assert.h"
char* stringDuplicator(char* s, int times) {
        assert(!s);
        assert(times > 0);
        int LEN = strlen(s);
        char* out = malloc(LEN*times);
        assert(out);
        for (int i=0; i<=times; i++) {
                out = out + LEN;
                strcpy(out,s);
        }
        return out;
}
```

We assume that the input is valid and not NULL so it should be without the exclamation point.

Variable name should be more informative.

Curly brackets at the beginning (and at the end) of a function should be in a separate line.

We should also allocate 1 byte for the char '\0' at the end of the string.

Variable name should be in lower case letters.

The condition should be: i<times.

We can't assume that the allocation succeeded, so we need to add a return value of NULL here. Otherwise, could be an undefined operation in strcmp.

This line should be just after the call to strcpy.

The function should return pointer to the beginning of the duplicated string, but because of the assignment in the for loop it doesn't.

2.1.2. The fixed code-

```
#include <stdlib.h>
#include <string.h>
#include <assert.h>
char* stringDuplicator(char* str, int times)
{
        assert(str);
        assert(times > 0);
        int len = strlen(str);
        char* out = malloc(len*times + 1);
        if (out == NULL) {
                return NULL;
        }
        for (int i=0; i<times; i++) {
                strcpy(out +i*len,str);
        }
        return out;
}
```

Yellow: Code convention errors.
Red: Code errors.