

### Hw1 dry:

#### 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;
}
```

#### Prograrmming Errors:

- 1.#include with "" instead of <>
- 2.assert(out) instead of checking if the dynamic allocation failed and returning NULL.
- 3.string created in malloc isn't long enough. we need space for closing the string with '/0', so we should add +1 to the length of the string we are creating.
- 4.<= in for loop instead of < making an access to unknown place in memory.
- 5.out is incremented and when returning it returns the end of the string, so now out points at the end of the string.
- 6.the order between incrementing the pointer and strcpy isn't correct, the first hello won't be copied to the new string.
- 7.assert (!s) isn't correct.

#### Convention Errors:

- 1.LEN instead of len
- 2.function name stringDuplicator instead of duplicateString
- 3.for loop doesn't contain spaces
- 4.duplicate assert call

Fixed code:

```
#include <stdlib.h>
#include <string.h>
#include <assert.h>

char* duplicateString(char* s, int times){
    assert(s != NULL && times > 0);
    int len = strlen(s);
    char* out = malloc(sizeof(char)*(len*times + 1));
    if(out == NULL)
    {
        Return NULL;
    }
    char* temp = out;
    for (int i=0; i < times; i++)
    {
        strcpy(temp,s);
        temp += len;
    }
    return out;
}
```