#include **<stdlib.h>**#include **<string.h>**#include **<assert.h>**#include **<stdio.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;  
} \*/****/\* Conventions mistakes:  
 \* 1. LEN written in capital letters. (line 8)***

***\* 2. there is no indentation after the lines that come after the "for" loop. (line 12,13)***

***\* 3. should name the function as a verb, like: "stringDuplicate" or "duplicateString". (line 5)***

***\* 4. Should set a more proper and meaningful name to the parameter out. (line 10)***

***\* 5. there is no comments of the code (what the code does etc.).***

***\*/  
  
 /\* Programming mistakes:  
 \* 1. not added enough space in the malloc allocation for the "/0" char needs to add +1 at the end. (line 9)***

***\* 2. need to assert "s", not "!s" - !s will assert the input is NULL. (line 7)***

***\* 3. you cant assert "out" because its allocated so you must check if its null or not. (line 10)***

***\* 4. "out" pointer inside the for loop starts with the second duplicated word (it will print the word times-1 times). (line 12)***

***\* 5. Pointer out is updated inside the loop so it will return NULL instead of the first letter in the char\*. (line 15)***

***\*/***

***// corrected code:***

#include **<stdlib.h>**#include **<string.h>**#include **<assert.h>**#include **<stdio.h>**

*//duplicates a string the number of times as the user asks* **char** \*duplicateString(**char** \*s, **int** times) {  
 **assert**(s);  
 **assert**(times > 0);  
 **int** len = strlen(s);  
 **char** \*duplicated\_string = malloc(len \* times + 1);  
 **if** (!out) {  
 **return NULL**;  
 }  
 **char**\* tmp\_ptr = duplicated\_string;   
 **for** (**int** i = 0; i < times; i++) {  
 strcpy(duplicated\_string, s);  
 duplicated\_string = duplicated\_string + len;  
 }  
 **return** tmp\_ptr; *//pointer to the beginning of the duplicated string.* }