

חלק יבש

```
1.#include "stdlib.h"
2.#include "string.h"
3.#include "assert.h"
4.char* stringDuplicator(char* s, int times){
5.assert(!s);
6.assert(times > 0);
7.int LEN = strlen(s);
8.char* out = malloc(LEN*times);
9.assert(out);
10.for (int i=0; i<=times; i++){
11.out = out + LEN;
12.strcpy(out,s);
13. }
14.return out;
15.}
```

שגיאות קונבנציה –

1. בשורה מס' 7 LEN הוא משתנה בעל אותיות גדולות ולפי הקונבנציה בקורס אסור לכתוב משתנים שכל האותיות שלהם גדולות.
2. כלולאה המתחילה בשורה 10 ומסתיימת בשורה 13 אין הזחה בתוך הלולאה וזו שגיאת קונבנציה בקורס.

שגיאות תכנות –

1. מאקרו ה include בשורות מס' 1-3 מכיל ספריות של המערכת עם "" ולא <.> צריך לשים < > כי אם יש לנו קובץ בעל אותו שם של אחת הספריות הללו הפרי פרוססור יכלול אותו ולא את הספרייה שרצינו
2. בשורה מס' 5 התוכנית תמשיך לפעול רק אם s=NULL וכמובן שאנחנו רוצים שהתוכנית תמשיך לרוץ רק אם s לא NULL.
3. בשורה מס' 8 אנחנו מקצים מקום בזיכרון שגודלו בדיוק כגודל הstring עם זאת אנחנו לא יודעים מתי הstring נגמר כי צריך להקצות מקום בזיכרון בשביל לדעת מתי הstring נגמר.
4. בשורה מס' 10 הלולאה רצה times+1 פעמים ולא times פעמים כפי שצריך
5. בשורות מס' 11-12 הפוינטר של out אינו יתחיל מהחלק הראשון של הstring ולכן ידלג על כמות מקומות בזיכרון בגודל len.
6. בשורה מס' 14 אנחנו מחזירים את out כאשר out לא מצביע למקום הראשון ב string החדש

הקוד המתוקן –

```
#include <stdlib.h>
#include <string.h>
#include <assert.h>
char* stringDuplicator(char* s, int times)
{
    assert(s); //check nullity
    assert(times > 0); //check that param is legal
    int len = strlen(s);
    char* out = malloc(len*times+1); //make space for null terminator
    assert(out);
    for (int i=0; i<times; i++){ //loop through 0 to times
        strcpy(out,s);
        out = out + len;
    }
    out[len*times]='\0'; //put null terminator
    return (out-len*(times)); //move to start
}
```