

```
#include "stdlib.h"
#include "string.h"
#include "assert.h"

char* stringDuplicator(char* s, int times) {
    ① assert(!s);
    ② assert(times > 0);
    int LEN = strlen(s);
    char* out = malloc(LEN*times);
    assert(out);
    for (int i=0; i<=times; i++) {
        out = out + LEN;
    ③ strcpy(out, s);
    }
    return out;
}
```

שגיאות קונבנצייה:

- לא רצוי לkür string ל-d, ניתן לkür אבל ל-str.
- שמות משתנים יש לכתוב באותיות קטנות

שגיאות תכונות:

- לפונקציה assert נדרש ערך בוליאני כלומר במקומות להכניס ?! נדרש להכניס NULL != !
בנוסף, שורה זאת מיותרת משום שנאמר שניית להניח את שהקלט נכון
- לפונקציה malloc נדרש את כמות הביטים לא כמות הchar'ים כלומר נדרש להכניס * times * sizeof(char)
- הfonקציה assert לא תחזיר NULL כנדרש אם malloc לא עבד וכאמור ב-1 יש להכניס לתוכה ערך בוליאני.
במקום בדיקה של assert יש לבדוק שNULL!=! out ובמקרה שהוא כן שווה NULL יש לשים זאת כערך החזרה של הfonקציה. ניתן לעשות זאת בעזרת if(out == NULL) return NULL;

```
if(out ==NULL)
{
    return NULL
}
```

- הלואה רצה פעם אחת יותר מיד, ניתן לתקן זאת על ידי להתחיל את ? ב-1 או על ידי שינוי התנאי עצירה ל- times < ?
- יש ניסיון חיבור בין int to string. ניתן להוריד שורה זאת מכיוון שאין לה מטרה בקוד.
- יש שימוש בfonקציית strcpy שפשט מעתיקה את string ליעד במקום strcat שמוסיפה את string למה שכבר קיים ביעד.

 לא טכנית שגיאה אבל גם במקרה כללי מיותר - זהו בדיקת תקינות הקלט, שכמובן לפי האדרות התרגיל מיותרת.

מואין הוגדר נספח
בנוסף לשליטה

הפלטת ה

```

1 #include "stdio.h"
2 #include "stdlib.h"
3 #include "string.h"
4 //#include "assert.h"
5
6 char* stringDuplicator(char* str, int times)
7 {
8     //no need to check if input is valid, but can be nice:
9     //assert(str != NULL);
10    //assert(times > 0);
11    int len = strlen(str);
12    char* out = malloc(len * times * sizeof (char));
13    if(out == NULL)
14    {
15        return NULL;
16    }
17    for (int i = 0; i < times; ++i)
18    {
19        strcat(out, str);
20    }
21    return out;
22 }
23
24 int main()
25 {
26     char* str = "hello";
27     printf("%s", stringDuplicator( str, 3));
28     return 0;
29 }
30

```