

## שגיאות תכנות

\* `assert` עוזר את החתית כיצר הביטוי בו הוא ביטוי שקר, ולכן - כל מקרה שבו תישלח מחרוזת שאינה `NULL` והחתית תקרוס אחי. ה `assert(!s)`.  
 לה הפך מורכב, כי חציס פלוצא שהמחרוזת לא ריקה.  
 אם היה צריך שהפוס `assert(s) !`.

\* ה `malloc` לא הנפילו `sizeof(*out)`, מכרז המקיים הדפול אכן 1 אבל לא היה ניתן להעביר את הקוד למערכת שבו `char` תופס 2 בתים עפחות / במקרה שירצו עשנות את הכיפוס של המילוק `char*`.  
 נן הקוד לא נ"צ.

\* `out = out + LEN` ← אכן עניק? לה פשוט  
 לא נכון להוסיף סרטיו, ובכח שאין `int`  
`(LEN)`

⚠️ הַפִּדְיוֹן (for(int i=0; i<=time; i++)) מְתִיקָה

time+1 פְּעִמִּים, לֹא לֹא  $1 = \text{time}$  הַפִּדְיוֹן תִּתְקַ"פּ

פְּעִמִּים, כִּי -  $i = 0, 1$  וְיִזְכַּר הַפִּדְיוֹן תִּתְקַצֵּץ אֶת

פְּעִמִּים מְרֻבָּעִים.

⚠️ strcpy(out, s) : הַפִּדְיוֹן הַזֶּה חֲצִיךְ כֹּל

פֶּעַם אֶחָד מִקֶּדֶם אֶת s (אֲחֵרָה לֹא out),

אֲחֵרָה אֶת הַפִּדְיוֹן הַזֶּה. צִיכָה אֶת הַפִּדְיוֹן

מִמֶּנּוּ מִחִלּוֹ הַפִּדְיוֹן.

## לשיווא קונבנציה:

⊛ שמות משתנים חייבים להיות באותיות קטנות

בארז וואו זאלה בתחילת כל מילה שיונה העונה.

בסוף ציה העצור: `int LEN = strlen(s);`

ציה עצור  
↑  
`len`

⊛ על השמות להיות משמעותיים ולא מקוצרים וזק

הל `char* stringDuplicator (char* s, .....)`  
↑  
לא משמעותי