

תרגיל בית 1

מת"מ אביב 23

2.1 סעיף א'

מצאו 6 שגיאות תכנות ו-2 שגיאות קונבנציה¹ (code convention) בפונקציה הבאה. מטרת הפונקציה היא לשכפל מספר פעמים את המחרוזת המתקבלת לתוך מחרוזת חדשה. למשל, הקריאה `stringDuplicator("Hello",3)` תחזיר את המחרוזת "HelloHelloHello". במקרה של שגיאה בריצת הפונקציה, הפונקציה תחזיר NULL. מותר להניח שהקלט תקין, כלומר אין צורך בבדיקה תקינות קלט כפי שלמדתם בתרגולים.

```
#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;
}
```

שגיאות קונבנציה:

1: המשתנה LEN מסומן באותיות גדולות למרות שהוא אינו מוגדר ע"י `#define` אלא משתנה ולכן אמור להיות `camelCase` (כלומר len)

2: בלולאת ה `for` לא נעשה שימוש נכון בהזחות, והתכנים שלה באותה רמת הזחות כמוה.

שגיאות קוד:

1: בשורה `assert(!s)` אמור להיות כתוב `assert(s)`. זהו מכיוון שמקרו `assert` עוצר את התכנה במידה והוא מקבל משהו עם ערך לוגי שקר, בפרט פוינטר שמצביע ל NULL, שזה המקרה שאנחנו נרצה לעצור ולכן לא נצטרך !.

2: בלולאת ה `for` הלולאה עוברת מ 0 עד `times` כולל מה שיגרום להעתקה של `times+1` עותקים של `s` ל `out`. (ולשגיאת segmentation מכיוון שלא הקצנו כמות מתאימה של זיכרון.)

3: פקודות ה `include` בתחילת הקובץ משתמשות בסוגרי " במקום סוגרי > למרות שהם של ספריות סטנדרטיות

4: לא נעשה `include` לספריית `string.h` למרות שנעשה שימוש בפונקציות המוגדרות בה `strlen` ו `strcpy`

5: בסיום הפונקציה מוחזר ה `out` האחרון בלולאה, שמצביע לעותק האחרון של המילה.

6: בלולאת ה `for` נעשה קודם כל שינוי באינדקס של `out` ורק לאחר מכן נעשה העתקה ל `out`, ולכן אנחנו מדלגים על המקום לעותק הראשון ב `out`, ולכן מבזבזים את הזיכרון הזה, וגשים לזיכרון שלא הקצנו באיטרציה האחרונה.

2.2 סעיף ב'

כתבו גרסה מתוקנת של הפונקציה.

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

char* stringDuplicator (char *s, int times) {
    assert (s);
    assert (times > 0);
    int len = strlen(s);
    char *out = malloc(len * times);
    char *start = out;
    assert (out);
    for (int i = 0; i < times; i++) {
        strcpy (out, s);
        out = out + len;
    }
    return start;
}
```