מת"מ – 1 תרגיל יבש מספק

איתי ברקוביץ 316088970

איל שטיין 208622142

טעויות בכתיבת הקוד:

- בשורה הראשונה של מימוש הפונקציה אנו נתקלים בביטוי **assert(!s)**, מטרת הביטוי היא לבדוק האם המחרוזת s ריקה, אמנם עפ"י המימוש הנוכחי בהינתן והמחרוזת לא ריקה התוכנית תעצור ותשלח הודעת שגיאה ל-stderr. לעומת זאת, הדרישה הייתה להחזיר NULL

- בהקצאת הזיכרון malloc(LEN*times), נשים לב כי LEN מכילה את אורך המחרוזת בלבד בלי מקום ל\0 בשביל לסמן את סוף המחרוזת האחרונה. לכן צריך לכתוב malloc(LEN*times + 1).
הערה: מכיוון ש1==(sizeof(char אין צורך לכתוב אותו בתוך הקצאת הזיכרון, אך זה תכנות לא נכון כי אם נרצה להחליף ממחרוזות של אותיות למערכים של מספרים ייתכן כי לא נשים לב להבדל בהקצאת הזיכרון.

- בשורה החמישית במימוש הפונקציה נכתב assert(out);. זוהי טעות בקוד כי דרישת התרגיל הייתה להחזיר NULL בעוד assert יקרא לפונקציה abort() במקרה של בעיה בריצת התוכנית ולא יחזיר NULL.

- בלולאת הfor, מתבצעות times+1 איטרציות כאשר בפועל יש הדרישה הייתה לשכפל times פעמים.

- בתוך לולאת הfor, סדר השורות צריך להתחלף כדי שנוכל להעתיק את התוכן מs לout ואז לקדם את המצביע למיקום בו צריך להתחיל השרשור הבא.

- Return out; בסוף מחזיר את המצביע לסוף המערך מכיוון שבכל איטרציה קידמנו את out מרחק של LEN בכמות של times+1 פעמים. לעומת זאת הדרישה הייתה להחזיר מצביע לתחילת המערך.

הערה: בניגוד למה שציפינו, מותר לכתוב #include "stdlib.h". לפי סטנדטרט C שעודכן בשנת 2018, פרק 6, תת פרק 10, פסקה 2, נכתב כי ביטוי אשר נכתב בין סימן " ובין סימן " ייבדק תלוי מימוש של קומפיילר (לרוב קודם בתיקיית המשתמש, current directory) ואם לא יימצא אז נתייחס אליו כאילו נכתב בין < ובין >.

קונבנציה:

- בתוך מימוש הפונקציה StringDuplicator אין שימוש בהזחות תקינות בקוד.

- המשתנה המקומי LEN צריך להיכתב באותיות קטנות.

- קריאה למחרוזת בשם s אינה עומדת בקונבנציה כי למחרוזת מותר לקרוא str (ללא קיצורים אחרים).

והנה הקוד כתוב בצורה תקינה:

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

char* stringDuplicator(char* str, int times){

        /*
        We're allowed to assume that the input is valid, but we can't
call
        strlen() on NULL,
        and if we pass a negative value to malloc it will get casted
to
        unsigned and get very big.
        Therefore, we will check just in case:
        */
        if (!str || times <= 0){
                return NULL;
        }
        int len = strlen(str);
        char* out = malloc(sizeof(char)*(len*times + 1));
        if (!out){
                return NULL;
        }
        char* startOfString = out;
        for (int i=0; i<times; i++){
                strcpy(out, str);
                out = out + len;
        }
        return startOfString;
}
```