

Hw1 חלק יבש

1. כשעושים include ל ספריות הקיימות (string, stdlib וכו') משתמשים ב<> ולא בגרשיים וכך preprocessor יודע אם לחפש קודם את הספריות הקיימות בקומפילר או תחילה לחפש headers וספריות שאנחנו בנינו ונמצאים באותה תיקיה. לכן צריך להחליף את הגרשיים ב<> ב3 השורות הראשונות בקוד. (קונבנציה)
2. בהגדרת הפרמטרים לפונקציה השם של הפרמטר למחרוזת הקלט 's' אינו משמעותי ולא ניתן להבין ממנו מה אנחנו צריכים להכניס לפונקציה. לכן נשתמש בstr עבור string (שגיאת קונבנציה)
3. כשעושים הנחות באמצעות assert אנחנו רושמים למה אנחנו מצפים. לכן בהנחה שהקלט תקין נרשום assert(s) ולא assert(!s). (שגיאת תכנות)
4. עבור שמות של משתנים (כמו LEN) נשתמש באותיות קטנות בלבד. (שגיאת קונבנציה)
5. כשעושים הקצאה דינמית למחרוזת הפלט אנחנו מכפילים את מספר הפעמים שנרצה לשכפל עם אורך המילה אבל צריכים להוסיף עוד בייט עבור "0" שמציין את סוף המחרוזת. (שגיאת תכנות)
6. השימוש ב assert "מוגבל" להנחות של המתכנת בקוד כדי לגלות באגים ושגיאות שאינן אמורות לקרות לעולם ונרצה למצוא אותן לפני שחרור התוכנה. לא ניתן להניח שהקצאה דינמית תמיד מצליחה ושגיאה כזו לא תלויה בקוד שלנו ועלולה לעיתים להתקבל ולעיתים לא. לכן אנחנו צריכים לבדוק אם ההקצאה תקינה באמצעות if למשל ולהחזיר שגיאה במידה וההקצאה לא תקינה. לרוב בשחרור התוכנה הנחות assert לא נבדקות, ולכן מלבד שימוש לא נכון בassert, לא תתבצע כלל בדיקה. (שגיאת תכנות)
7. לא הוחזר NULL במידה וההקצאה הדינמית לא הצליחה. (שגיאת תכנות)
8. בבלוק של לולאת for נצטרך להוסיף הזחות. (שגיאת קונבנציה)
9. תנאי העצירה שגוי, הלולאה מתבצעת פעם נוספת (שגיאת תכנות)
10. לולאת for בנויה לא נכון וההעתקה לא מתבצעת כהלכה. (שגיאת תכנות)
11. בתהליך ההעתקה מקדמים את המצביע למחרוזת הפלט 'out' שוב ושוב באורך LEN אבל לא שומרים את המצביע לתחילת המחרוזת ולכן בסוף הפונקציה המצביע הנשלח הוא לא מצביע לתחילת מחרוזת הפלט. (שגיאת תכנות)

הערה :

1. בפונקציה stringDuplicator אנחנו עושים שכפול למחרוזת שקיבלנו אך צריכים לדאוג שלא נשנה אותה. לכן נצטרך להגדיר את הפרמטר s כקבוע. ניתן לראות שהשימוש ב's' בפונקציות המוכרות לנו (strcpy וכו') יפגעו במחרוזת לכן במקרה זה זו תהיה שגיאת קונבנציה כי נרצה שהמשתמש ידע שהוא מוגן כאשר משתמש בפונקציה. (שגיאת קונבנציה)

```

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

char* stringDuplicator(const char* str, int times) {
    assert(str);
    assert(times > 0);
    int len = strlen(str);
    char* out = malloc (len*times + 1) ;
    if (out == NULL) {
        return NULL;
    }
    char* tmp_out = out;
    For (int i = 0 ; i < times ; i++) {
        If(!strcpy(tmp_out, str)) { //strcpy failed
            free(out);
            return NULL;
        }
        tmp_out = tmp_out + len;
    }
    return out;
}

```