

חלק יבש – זיהוי שגיאות בקוד

שרה גריפית + ליאור בר יוסף
207022443 + 341312304

חלק א' – זיהוי השגיאות:

שגיאות קונבנציה:

1. **שורות 12-13:** התוכן של לולאת ה-for צריך להיות עם אינדנטציה (indent) אחת ימינה. כלומר, אינדנטציה אחת ימינה מעבר לשורה של פקודת ה-for (שנמצא בשורה 11).
2. **שורה 7:** שם המשתנה LEN צריך להיות בlowercase כיוון שאינו משתנה סטטי / קבוע. משתנים זמניים צריכים להיות בcamelCase.

שגיאות תכנות:

1. **שורות 1-3:**
 - a. מצב נוכחי: כל שלושת ה-#include ימים מכילים את הספיריות במרכאות ("").
 - b. תיקון: כל שלושת ה-#include ימים צריכים להיות בתוך משולשים <>, זהו כדי שהמערכת תחפש את הספיריות בניתובים ידועים וקונבנציונליים, כמו שמתאים עבור הספיריות המוכרות הללו.
2. **שורה 6 ו-10 שימוש לא נכון בassert:**
 - a. שורה 6 - מצב נוכחי: הassert הראשון מוודא שהמחרוזת מקלט הפונקציה שווה לNULL.
 - i. תיקון: הassert הראשון צריך לוודא שהמחרוזת מקלט הפונקציה לא שווה ל-NULL.
3. **שורה 9:**
 - a. מצב נוכחי: גודל הקצאת זיכרון בפקודת malloc לא נכון עבור מחרוזת.
 - b. תיקון: צריך להוסיף את הערך +1 להקצאת הזיכרון על מנת להכניס את התו '\0' בסוף המחרוזת החדשה, אחרת תהיה זליגת זיכרון.
4. **שורה 11:**
 - a. מצב נוכחי: הספירה של לולאת ה-for מתחילה מ-0 ולכן המילה מועתקת times+1 פעמים.
 - b. תיקון: צריך לשנות את הלולאה כך שהתנאי יהיה קטן ממש, כלומר השורה צריכה להיות:
for (int i=0; i<times; i++) {
5. **שורות 12-13:**
 - a. מצב נוכחי: ההעתקה של המילים מתחילה מאמצע המחרוזת החדשה, ולא בתחילתה.
 - b. תיקון: צריך להחליף בין השורות בתוך לולאת ה-for (שורות 12 ו-13), כדי שקודם נעתיק את המחרוזת בפעם הראשונה ורק אז נזיז את pointer למיקום של תחילת העותק הבא של המחרוזת.
6. **שורה 15:**
 - a. מצב נוכחי: הפוינטר שמוחזר מהפונקציה פונה לסוף המחרוזת.
 - b. תיקון: צריך להחזיר את הפוינטר כאשר הוא פונה לתחילת המחרוזת. נעשה זאת על ידי יצירת עותק חדש של הפוינטר למחרוזת והזזתו בעת העתקת המילים, בעת שהפוינטר המקורי פונה עדיין לתחילת המחרוזת, והוא הערך שיוחזר מהפונקציה.

חלק ב' – הקוד המתוקן:

```
#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 + 1);
    if (!out) {
        return NULL;
    }
    char* current = out;
    for (int i=0; i<times; i++) {
        strcpy(current,s);
        current = current + len;
    }
    return out;
}
```