**חלק יבש – זיהוי שגיאות בקוד**

**שרה גריפית + ליאור בר יוסף**

**341312304 + 207022443**

**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:** 
   1. מצב נוכחי:כל שלושת ה- #include-ים מכילים את הספיריות במרכאות ("").
   2. תיקון:כל שלושת ה- #include-ים צריכים להיות בתוך משולשים צדדים (<>), זהו כדי שהמערכת תחפש את הסיפריות בניתובים ידועים וקונבנציונליים, כמו שמתאים עבור הסיפריות המוכרות הללו.

כלומר, צריך לרשום **#include <string.h>**

1. **שורה 6 ו10 שימוש לא נכון בassert:**
   1. שורה 6 - מצב נוכחי: הassert הראשון מוודא שהמחרוזת מקלט הפוקנציה שווה לNULL.
      1. תיקון: הassert הראשון צריך לוודא שהמחרוזת מקלט הפוקנציה לא שווה ל-NULL.

כלומר, צריך לרשום **assert(s);**

* 1. שורה 10 - מצב נוכחי: בדיקה של ההקצאה הדינאמית על ידי פקודת assert.
     1. תיקון: צריך לשנות את הפקודת assert בתנאי if, אשר מחזיר את הערך NULL במקרה שלא הוקצה זיכרון דינאמי. לא נכון להשתמש בassert כי assert מתבצע רק בזמן debugging, וצריך לבדוק את התנאי הזה תמיד.

1. **שורה 9:**
   1. מצב נוכחי: גודל הקצאת זיכרון בפקודת malloc לא נכון עבור מחרוזת.
   2. תיקון: צריך להוסיף את הערך +1 להקצאת הזיכרון על מנת להכניס את התו ‘\0’ בסוף המחזרות החדשה, אחרת תהיה זליגת זיכרון.
2. **שורה 11:**
   1. מצב נוכחי: הספירה של לולאת ה-for מתחילה מ-0 ולכן המילה מועתקת times+1 פעמים.
   2. תיקון: צריך לשנות את הלולאה כך שהתנאי יהיה קטן ממש, כלומר השורה צריכה להיות: **for (int i=0; i<times; i++) {**
3. **שורות 12-13:** 
   1. מצב נוכחי: ההעתקה של המילים מתחילה מאמצע המחרוזת החדשה, ולא בתחילתה.
   2. תיקון:צריך להחליף בין השורות בתוך לולאת ה-for (שורות 12 ו-13), כדי שקודם נעתיק את המחרוזת בפעם הראשונה ורק אז נזיז את הpointer למיקום של תחילת העותק הבא של המחרוזת.
4. **שורה 15:** 
   1. מצב נוכחי: הפוינטר שמוחזר מהפונקציה פונה לסוף המחרוזת.
   2. תיקון: צריך להחזיר את הפוינטר כאשר הוא פונה לתחילת המחרוזית. נעשה זאת על ידי יצירת עותק חדש של הפוינטר למחרזות 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;  
}