**חלק יבש – זיהוי שגיאות בקוד**

**חלק א' –**

שגיאות קונבנציה:

1. **שורות 12-13:** בתוך לולאת ה-for השורות צריכות להיות עם אינדנטציה (indent) אחת מעבר לשורה של פקודת ה-for (בשורה 11).
2. **שורה 7:** שם המשתנה LEN צריך להיות בlowercase כיוון שאינו משתנה סטטי / קבוע. משתנים זמניים צריכים להיות בcamelCase. כלומר, שם המשתנה צריך להיות בlowercase, ואם השם מורכב מכמה מילים אז האות הראשונה של כל מילה צריכה להיות בuppercase (לא כולל המילה הראשונה).

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:**
   1. מצב נוכחי: הassert הראשון מוודא שהמחרוזת מקלט הפוקנציה שווה לNULL.
   2. תיקון: הassert הראשון צריך לוודא שהמחרוזת מקלט הפוקנציה לא שווה ל-NULL.

כלומר, צריך לרשום **assert(s);**

1. **שורה 10:**
   1. מצב נוכחי: בדיקה של ההקצאה הדינאמית על ידי פקודת assert.
   2. תיקון: צריך לשנות את הפקודת assert בתנאי פשוט, אשר מחזיר את הערך NULL במקרה שלא הוקצה זיכרון דינאמי.

התיקון הזה נובע משתי סיבות: פקודת assert יותר מתאימה לוידאו ארגומנטים (כמו שיש בתחילת הפונקציה הזו), בדיקת ערכי חזרה וכו'... ופחות לוידאו הקצאות זיכרון. בנוסף לכך, דרישות התרגיל הן שיוחזר ערך NULL אם יש שגיאות בזמן ריצת הפונקציה, ולכן צריך להוציא את התנאי מהassert.

1. **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++) {**
2. **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.
3. **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' –**

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