**חלק יבש – זיהוי שגיאות בקוד**

**חלק א' –**

שגיאות קונבנציה:

1. בתוך לולאת ה-for (בשורות 12-13) צריך שתהיה אינדנטציה של השורות לאחר פתיחת הסוגריים!
2. שם המשתנה LEN היא בuppercase, כאשר צריך שיהיה בcamelCase. כלומר, 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:** כל שלושת ה- #include צריכים להיות בתוך משולשים צדדים (<>) ולא במרכאות (""). זהו כדי שהמערכת תחפש את הספריה לא רק בתיקייה הנוכחית אלא בניתוב הקונבנציונלי, כפי שמתאים לספריות מוכרות כאלו.

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

1. **שורה 6:** הassert הראשון מוודא כי המחרוזית בקלט הפונקציה שווה לNULL, והיא צריכה לוודא את ההפך, שהמחזרות לא שווה לNULL.
2. **שורה 10:** צריך לשנות את הפקודת assert בשורה 10 בתנאי פשוט, אשר מחזיר את הערך NULL במקרה שלא הוקצה זיכרון דינאמי. זהו בגלל דרישות התרגיל, אשר בו יחזר ערך NULL אם היו שגיאות בזמן ריצת הפונקציה. פקודת assert יותר מתאימה לוידאו ארגומנטים (כמו שיש בתחילת הפונקציה הזו), בדיקת ערכי חזרה וכו'... ופחות לוידאו הקצאות זיכרון.
3. **שורה 11:** צריך לשנות את הלולאה כך שהתנאי יהיה קטן ממש, כלומר להסיר את השווה באדום:

**for (int i=0; i<=times; i++){**

הספירה מתחילה מ-0 ולכן במצב הנוכחה המילה מועתקת times+1 פעמים, ולכן צריך להקטין את הריצה ב-1.

1. **שורות 12-13:** צריך להחליף בין השורות בתוך לולאת ה-for (שורות 12 ו-13), כדי שקודם נשים את המחרוזת ורק אז נזיז את הpointer למיקום של העותק הבא של המחרוזת.

במצב הנוכחי התווים הראשוניים "ריקים".

1. **שורה 15:** צריך להחזיר את הפוינטר לתחילת המחרוזית, אך במצב הנוכחי הוא מוחזר לסוף המחרוזת!

#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;  
}