חלק יבש

סעיף א:

שגיאות תכנות:

1. שורה 4:

```
assert(!s);
```

s הוא מצביע למערך המחזורות שהפונקציה קיבלה. בשורה זו אנו בעצם רוצים לבדוק שהמצביע הזה לא NULL- אבל assert יזרוק שגיאה כאשר מה שרשום בסוגריים **לא מתקיים-** בסוגריים רשום s! - כלומר אם s=NULL, הערך של NULL הוא 0, כלומר false ו"לא 0" אומר "לא false- כלומר true- כלומר מה שבתוך הסוגריים כן יתקיים, ואנו נמשיך הלאה בקוד, כאשר אנו רוצים בדיוק את ההיפך! לכן סימן הקריאה בשורה זו הוא שגיאת תכנות.

2. שורה 8:

```
char* out = malloc(LEN * times);
```

יש לשים לב שstrlen לא סופר את התו '\0' בסוף המחרוזת, לכן כאשר נקצה מקום למחרוזת המשוכפלת יש להקצות מקום כאורך המחרוזת המשוכפלת **ועוד 1**, על מנת שיהיה מקום לתו '\0'.

3. שורה 9:

```
assert(out);
```

קוד שנמצא בתוך assert לא יורץ כלל אם נשתמש בדגל NDEBUG - אסור לשים חישובים הכרחיים לקוד בתוך assert! לבדוק האם הקצאת הזיכרון הצליחה זוהי בדיקה חשובה לקוד, רצוי לא לשים אותה בתוך assert אלא בתוך if.

4. שורה 10:

```
for (int i = 0; i <= times; i++)
```

הלולאה מתחילה מ-i=0 וצריכה לרוץ times פעמים כדי לשכפל את המילה times פעמים, לכן בתנאי הלולאה צריך להופיע i<times ולא i<=times.

5. שורות 11-12:

```
out = out + LEN;
strcpy(out, s);
```

באיטרציה הראשונה אנו מקדמים את out לפני ששמנו בהתחלה שלו ערך כלשהו,לכן סדר הפעולות צריך להיות הפוך.
בנוסף, אנו מקדמים את out בכל איטרציה ובסוף מחזירים את out כאשר הוא מצביע ל-LEN מקומות לפני סוף המחרוזת במקום לתחילת המחרוזת.

6. שורות 1-3:

```
#include "stdlib.h"
#include "string.h"
#include "assert.h"
```

שם ה-header file צריך להיות תחום ב-<> ולא ב-"" -מפניי שאלו הם קבצי ספריה ולא קבצים שנמצאים בתיקיה של הקובץ שבו נכתבה בתכנית.
ישנם קומפיילרים שיודעים "להתמודד" עם זה אך ראוי לכלול קבצי ספריה עם <>.

שגיאות קונבנציה:

1. שורה 4:

```
char* stringDuplicator(char* s, int times) {
```

שמות פונקציות צריכות להיות מנוסחות כפעלים.

2. שורה 4:

```
char* stringDuplicator(char* s, int times) {
```

s אינו קיצור חוקי למשתנה מחרוזת.

3. שורה 7:

```
int LEN = strlen(s);
```

שם של משתנה צריך להיות באותיות קטנות בלבד.

4. שורות 11-12:

```
strcpy(temp, s);
temp = temp + LEN;
```

שתי שורות אלו נמצאות בבלוק של לולאת for, אך אינן שומרות על מבנה קוד מפניי שאין הזחה.

סעיף ב:

גרסה מתוקנת של הפונקציה:

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

char* duplicatString(char* str, int times) {
	assert(str);
	assert(times > 0);
	int len = strlen(str);
	char* out = malloc((len * times) +1);

	if (out==NULL)
	{
		printf("Dynamic allocation failed");
		return NULL;
	}

	char* temp = out;
	for (int i = 0; i < times; i++)
	{
		strcpy(temp, str);
		temp = temp + len;
	}
	return out;
}
```