

## 2.1

הנה הקוד המקורי שקיבלנו:

```
#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);
    assert(out);
    for(int i=0; i<=times; i++) {
        out = out + LEN;
        strcpy(out,s);
    }
    return out;
}
```

### שגיאות תכנות:

1. בלולאת ה-for התנאי שאמור להיות הוא  $i < times$  ולא  $i \leq times$  כי זה יצור מצב של כתיבה פעם אחת יותר מידי.
2. ה-assert צריך להיות `assert(s)` ולא `assert(!s)` כדי לבדוק שהמחרוזת לא ריקה.
3. הפונקציה מחזירה את out אחריי שהיא מקדמת אותו והיא צריכה להחזיר את ההתחלה של המחרוזת.
4. הפונקציה מקדמת את הכתובת שאליה כותבים לפני שהיא כותבת בה, כלומר היא לא כותבת בכתובת המקורית, והיא כותבת פעם אחד מקדימה יותר ממה שמצופה.
5. כאשר אנו מצרפים ספריות מוכרות שנמצאות במיקומים קבועים, נצרף אותם כך `#include < assert.h >` למשל עם סוגריים משולשים כיוון שאחרת הם יחפשו קודם בתיקיות של הפרויקט בלי סיבה.
6. בלולאת ה-for התנאי שאמור להיות הוא  $i < times$  ולא  $i \leq times$  כי זה יצור מצב של גלישה מהזיכרון שעשינו לו malloc

### שגיאות קונבנציה:

1. השם LEN לא מתאים לint לפי הקונבנציות כי הוא צריך להיות באותיות קטנות.
2. שם הפונקציה הינו stringDuplicator שזה שם עצם. נצטרך לשנות את זה לשם פועל לפי הקונבנציה.

## 2.2

הנה גרסה מתוקנת של הקוד:

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

char* duplicateString(char* s, int times) {
    assert(s);
    assert(times > 0);
    int len = strlen(s);
    char* out = malloc(len * times);
    assert(out);

    for (int i = 0; i < times; i++) {
        strcpy(out + i * len, s);
    }

    return out;
}
```