

חלק יבש

סעיף א'

```
#include "stdlib.h"
#include "string.h"    < > משתמשים ב
#include "assert.h"

char* stringDuplicator(char* s, int times){
    משתמשים בשמות עם משמעות כמו str/string
    assert(!s);
    צריכים למחוק את הNOT כי ASSERT מחזירה כאשר מקבלת FALSE.
    assert(times > 0);
    int LEN = strlen(s);          שמות משתנים באותיות קטנות
    char* out = malloc(LEN*times+1);  חסר מקום לתו הסיומת
    assert(out);
    משתמשים בif כדי להחזיר NULL ולא לגרום לחריגה (תוך כדי שחרור זכרון)

    for (int i=0; i<=times; i++){
        הנכון בלי השווה כדי לא לקבל איטרציה נוספה
        out = out + LEN;          קודם מעתיקים ואז עוברים לחלק הבא
        strcpy(out, s);
    }
    חסרה הוספה של תו הסיומת למחרוזת
    *out = '\0';
    צריכים להחזיר את תחילת המחרוזת
    out -= times*len;
    return out;
}
```

סעיף ב'

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

char* stringDuplicator(char* str, int times){
    assert(str);
    assert(times > 0);
    int len = strlen(str);
    char* out = malloc(len*times+1);
    if (!out){
        free(out);
        return NULL;
    }
    for (int i=0; i< times; i++){
        strcpy(out, str);
        out = out + len;
    }
    *out = '\0';
    out -= times*len;
    return out;
}
```

2.2 מיזוג רשימות מקושרות ממוינות:

```
Node mergeSortedList(Node list1, Node list2){
    if(list1 == NULL || list2 == NULL){
        return NULL;
    }
    if(!isListSorted(list1) || !isListSorted(list2)){
        return NULL;
    }
    Node merged_list;
    if (list1 -> val < list2 -> val){
        merged_list = list1;
        list1 = list1 -> next;
    }else{
        merged_list = list2;
        list2 = list2 -> next;
    }
    Node head = merged_list;

    while (list1 != NULL && list2 != NULL){
        if (list1 -> val < list2 -> val){
            head -> next = list1;
            list1 = list1 -> next;
        } else {
            head -> next = list2;
            list2 = list2 -> next;
        }
        head = head -> next;
    }

    if (list1 != NULL){
        head -> next = list1;
    }

    if (list2 != NULL){
        head -> next = list2;
    }

    return merged_list;
}
```