

שאלה 1 - מיזוג רשימות מקושרות ממוינות:

```
Node mergeSortedList(Node list1, Node list2, ErrorCode* error_code) {
    int length1 = getListLength(list1);
    int length2 = getListLength(list2);
    if (length1 == 0 || length2 == 0) {
        *error_code = NULL_ARGUMENT;
        return NULL;
    }
    if (!isListSorted(list1) || !isListSorted(list2)) {
        *error_code = UNSORTED_LIST;
        return NULL;
    }
    int merged_list_length = length1 + length2;
    Node merged_list = malloc(sizeof(*merged_list));
    if (merged_list == NULL) {
        *error_code = MEMORY_ERROR;
        return NULL;
    }
    Node merged_list_iterator = merged_list;
    for (int i=0; i<merged_list_length; i++) {
        if (list1->x >= list2->x) {
            merged_list_iterator->x = list2->x;
            list2 = list2->next;
        }
        else {
            merged_list_iterator->x = list1->x;
            list1 = list1->next;
        }
        if (i == merged_list_length-1) {
            *error_code = SUCCESS;
            return merged_list;
        }
        Node merged_list_next = malloc(sizeof(*merged_list));
        if (merged_list_next == NULL) {
            *error_code = MEMORY_ERROR;
            return NULL;
        }
        merged_list_iterator->next = merged_list_next;
        merged_list_iterator = merged_list_next;
    }
    return NULL;
}
```

שאלה 2 - מציאת שגיאות :

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

char* foo(char* str, int* x) {
    char* str2;
    int i;
    x = strlen(str);
    str2 = malloc(*x);
    for (i = 0; i < *x; i++)
        str2[i] = str[*x - i];
    if (*x % 2 == 0) {
        printf("%s", str);
    }
    if (*x % 2 != 0)
    {
        printf("%s", str2);
    }
    return str2;
}
```

• שגיאות נכונות במימוש הפונקציה :

1. שורה 10 :
שגיאה: `str2[i] = str[*x-i]`
תיקון: `str2[i] = str[*x-i-1]`
2. שורה 12 :
שגיאה: `printf("%s", str)`
תיקון: `printf("%s", str2)`
3. שורה 16 :
שגיאה: `printf("%s", str2)`
תיקון: `printf("%s", str)`

• חריגות מכללי תכנות נכון :

1. הפונקציה קיבלה את הארגומנט `char* str` ולא נבדק אם הוא מצביע ל `NULL`.

תיקון: נוסף את הבדיקה בתחילת הפונקציה:

```
if(str == NULL)
{
    return NULL;
}
```

2. הפונקציה קיבלה את הארגומנט `int* x` ולא נבדק אם הוא מצביע ל `NULL`.

תיקון: נוסף את הבדיקה בתחילת הפונקציה:

```
if(x == NULL)
{
    return NULL;
}
```

```
}
```

3. אחרי הקצאת זיכרון למחרוזת str2 כלומר שורה 5, לא בדקו אם הוא מצביע ל NULL.

תיקון: נוסיף את הבדיקה אחרי ההקצאה:

```
if(str2 == NULL)
{
    return NULL;
}
```

4. שורה 8, הקצאת מקום בזיכרון ע"י שימוש ב malloc בלי בדיקה אם ההקצאה הצליחה.

תיקון: נוסיף את הבדיקה אחרי ההקצאה:

```
if(str2 == NULL)
{
    return NULL;
}
```

5. לא שחררו את הזיכרון שהוקצה ע"י malloc .

תיקון: נשחרר את המקום הזה בזיכרון בסוף הפונקציה לפני החזרת הערך המבוקש, כלומר נוסיף:

free(str2);

ובכך, מצאנו ותיקנו 8 שגיאות בקוד, לפחות 3 בכל סוג.