

חלק יבש – זיהוי שגיאות בקוד

סעיף א -

6 שגיאות תכנות בקוד:

- (1). `#include` – שימוש לא נכון ב `#include`
מתקיים כי `"file_name"` `#include` הוא עבור תיקייה שנכתבה על ידי המשתמש, לעומת, `#include <file_name>` שבה משתמשים עבור תיקייה מהספרייה.
ההבדל בין שני הסוגים הוא במיקום שבו ה- `preprocessor` מחפש את הקובץ שייכלל בקוד.
 - (2). הקצאת זכרון דינמי ללא בדיקתו -
לאחר שימוש ב `malloc` אנו חייבים לבדוק אם ההקצאה הצליחה, לעומת זאת משתמשים בקוד ב `assert(out)`, שבתוכנית הרצה ללא `debug`, הפקודה הנ"ל לא תעבוד ולכן עלולים להיווצר באגים.
 - (3). הקצאת זיכרון לא מספיק גדול -
אנו מבצעים `char* out = malloc (LEN*times)`.
נשים לב שלא יהיה מקום ל `(/0)` שמשמעותה סיום המחרוזת.
 - (4). לולאת `for` רצה יותר מידי פעמים:
`for(int i = 0 ; i <= times ; i++)`
כלומר הלולאה רצה `times+1` פעמים ולכן הלולאה לא תעבוד כנדרש.
 - (5). שימוש לא נכון ב `asset(!s)`
למעשה, אנו נקבל שגיאה מ `assert(!s)` במידה ו `s` בעלת כתובת חוקית. הכוונה פה היא לבדוק שהמחרוזת לא ריקה, ולכן זו שגיאה.
 - (6). שגיאה נוספת בתוך הלולאה, קודם מבצעים `out=out + len` ורק אז משתמשים ב `strcpy`.
כתוצאה מכך, החלק הראשון של הזכרון `out` באורך `len` לא נמצא בשימוש, וגם נבצע חריגה מהזכרון שהוקצה עבור `out`. בנוסף, אנו מחזירים בסוף את `out`, כלומר לא נחזיר את מיקום המצביע הנכון.
- 2 שגיאות קונבנציה בקוד:
- (1). `char* s` - לפי ה `Code conventions` שמופיעים באתר, הקיצור שמותר להשתמש בו עבור `string` הינו `str` ולא `s`. ולכן זוהי שגיאת קונבנציה
 - (2). `Int LEN` – שמות של משתנים צריכים להכיל אותיות קטנות בלבד ואות גדולה בתחילת כל מילה שאינה ראשונה. ולכן, המשתנה בצורתו התקינה תהיה `int len`, מאחר ומותר לעשות את הקיצור `len` עבור `length`.

סעיף ב -
גרסה מתוקנת של הקוד:

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

//This function returns the allocated array. Note that the
//caller is responsible to free it.
char* stringDuplicator(char* str, int times){
    assert(str);
    assert(times > 0);
    int len = strlen(str);
    char* out = (char*)malloc(sizeof(*out)*(len*times+1));
    if(out == NULL){
        printf("Dynamic allocation failed!\n");
        return NULL;
    }
    for (int i=0; i< times; i++){
        strcpy(out+i*len, str);
    }
    return out;
}
```