ilansh
302514401
Ilan Shamir

Appeal for ex4 c: (original README at the end of the appeal)

I would like to appeal my grade in ex 4 (0), which was caused by failed compilation of the file NRTest.c (caused by an error in the files MyNR.h and MyNR.c).
The error was that I defined the realFunction typedef as a function rather than a pointer to a function.
The code I wrote was:

```
typedef double realFunction(double);
```

The correct code is:

```
typedef double (*realFunction)(double);
```

This syntax obviously worked with the testers you provided us (otherwise I wouldn't have submitted it), but failed when trying to assign realFunction types to an array as performed in your tester (c does not accept functions as array members, only pointers to functions). I understand the mistake, but I would like to point out that this wasn't clear from the exercise description, which states (quote):

> בקובץ ה header)MyNR.h(עליכם להגדיר טיפוס-פונקציה בשם realFunction שיתאר
> פונקציות המקבלות ארגומנט מטיפוס double ומחזירות ערך מטיפוס double. טיפוס הפונקציה
> הזה יגדיר טיפוס של כל פונקציה שנרצה לחשב את נקודות ההתאפסות שלה.

According to given explanation, one can easily make this mistake and define the typedef as a function itself rather than a pointer to a function.

In addition, I accidentally wrote a different declaration in the cpp and the h file:

h file: double myNR(realFunction func, realFunction der, double start);
c file: double myNR(realFunction *func, realFunction *der, double start)
(the correct definition is in the h file without *)

I suppose this shows that I was somewhat confused by the requirement, but understood the general concept of using pointers to functions. Unfortunately, I didn't pay much attention to this file since the code is so short and all the testers passed (including those in the forum).

When taking off points for the fix, please take into consideration the fact that this was more of a syntax error rather than a logic error, and that this specific question is worth only 13 points.

Thanks,
Ilan.

(Fixed files – MyNR.c MyNR.h)

Original README:

1. a. func2, func3.
   b. func5.
   c. func6.
   d. func1.
   e. func2, func3.
   f. func7.
   g. func2, func3 (can be initialized in the array without a warning).
   e. func5.

2. The problem is that the malloc is called without a cast to (double*) so pointer arithmetic is not applied on the last '+1', meaning, the address sent to the function fillStudentGrades is incorrect because it is only byte apart from the start of the allocated area, while it should be 8 bytes apart (enough space for double).
The solution would be to use (double*) cast in front of malloc, applying pointer arithmetic
(we could also write sizeof(double) instead of +1).

Valgrind input: valgrind --leak-check=full HashInts 10

valgrind output:
==12038== HEAP SUMMARY:
==12038==   in use at exit: 0 bytes in 0 blocks
==12038==   total heap usage: 81 allocs, 81 frees, 1,300 bytes allocated
==12038==
==12038== All heap blocks were freed -- no leaks are possible
==12038==
==12038== For counts of detected and suppressed errors, rerun with: -v
==12038== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 4 from 4)