Lab #4.5

Summer 2024

Requirements

This is a bonus lab and is optional. If you have time after completing Lab 4, you can complete this lab for bonus points. If you don't have time or don't wish to implement the lab, no worries at all!

To complete the lab, all you have to do is implement jagged 2-dimensional smart arrays as we discussed in class. You must use the prototypes defined in the header file provided in the assignment on Canvas.

In addition to the implementation file (lab4.5.c), you should write your own main program to test things. To make sure you don't have memory leaks, you may want to try running your code with valgrind.

Here is an example of running my implementation with valgrind:

```
jimr@jimrsurfacepro9:~/CodeMizzou/demos/2050.04.5$ valgrind ./a.out
==1252== Memcheck, a memory error detector
==1252== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==1252== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==1252== Command: ./a.out
==1252==
*** Here comes test #1 ***
array[0][0]=0, array[0][1]=1,
array[1][0]=2, array[1][1]=3, array[1][2]=4, array[1][3]=5,
array[2][0]=6, array[2][1]=7, array[2][2]=8,
array[3][0]=9, array[3][1]=10, array[3][2]=11, array[3][3]=12, array[3][4]=13, array
[3][5]=14, array[3][6]=15,


*** Here comes test #2 ***
Jim Ries (Human) - 57, Laura Ries (Human) - 55, Abbie Ries (Human) - 25, Charlotte R
ies (Human) - 19, Cisco Ries (Canine) - 13, Murphy Ries (Canine) - 1,
Larry Ries (Human) - 55, Allison Ries (Human) - 14,
==1252==
==1252== HEAP SUMMARY:
==1252==     in use at exit: 0 bytes in 0 blocks
==1252==   total heap usage: 9 allocs, 9 frees, 7,344 bytes allocated
==1252==
==1252== All heap blocks were freed -- no leaks are possible
==1252==
==1252== For lists of detected and suppressed errors, rerun with: -s
==1252== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Note that my first test is a jagged array of type int with 4 rows and column sizes 2,4,3,7. The second test is a jagged array of type FamilyMember (a struct I defined in my main file) with 2 rows and column sizes 6,2.

Submission Information

Submit this assignment by using the mucsmake command.

Use the following submit command on Hellbender:

mucsmake <course> <assignment> <filename>

For example:

mucsmake 2050 lab4.5 lab4.5.c

Rubric: 4 points (bonus points; will be up to "4 out of 0")

1. AllocateJagged2DArray() is correct
   * 1 point
2. FreeJagged2DArray() is correct
   * 1 point
3. GetRowCount() is correct
   * 1 point
4. GetColCount() is correct
   * 1 point

Notice:

1. All of your lab submissions **must** include documentation in the form of code comments to receive full points. In addition, your program is expected to have a **comment header** at the top that includes your name, pawprint, the course you are taking, and the lab that you solved. You can refer to the Lab 0 document for an example of the comment header.
2. All of your lab submissions must compile under GCC using the -Wall and -Werror flags to be considered for a grade. These flags will automatically be applied if you use the compile command.
3. Do **NOT** change the given function prototype or anything else in the provided .h file.