

Author: Benjamin R. Olson
Date: April 13, 2014
Class: cs162 - Introduction to Computer Science II
Description: Project 1 Tests

Tests during implementation:

file: test1.cpp

configuration 1: ul corner

(where x = 0, gets neighbors of upper-left corner cell):

```
x4***|
67***|
*****|
```

test: gdb to navigate values in neighbors array

expected result: neighbors array is

```
{',', ',', ',', ',', '4', ',', '6', '7'}
```

configuration 2: ur corner

(where x = 4, gets neighbors of upper-right corner cell):

```
***3x|
***56|
*****|
```

test: for loop to output neighbors array

expected result: neighbors array is

```
{',', ',', ',', '3', ',', '5', '6', ','}
```

actual result:

```
{',', ',', ',', '3', ',', 'x', '*', ','}
```

error fixed:

sixth assignment: arr[width - 1] changed to...

arr[x + width - 1]

seventh assignment: arr[width] changed to...

arr[x + width]

configuration 3: top row

(where x = 3, gets neighbors of a cell in the top row, but not corner):

```
**3x4|
**567|
*****|
```

test: for loop to output neighbors array

expected result: neighbors array is

```
{',', ',', ',', '3', '4', '5', '6', '7'}
```

actual result:

```
{',', ',', ',', '3', '4', '*', ',', ','}
```

error fixed:

in last three statements, wrong indexes assigned and wrong arithmetic

configuration 4: bottom row

(where x = 11, gets neighbors of a cell in the bottom row, but not corner):

```
*****|
012**|
3x4**|
```

test: for loop to output neighbors array

expected result: neighbors array is

```
{'0', '1', '2', '3', '4', '', '', ''}
```

actual result:

```
{'2', '*', '*', '3', '4', '', '', ''}
```

error fixed:

in the first three assignments,

neighbors[y] = arr[x - (width-1) + y]; changed to

neighbors[y] = arr[x - (width+1) + y];

configuration 5: ll corner

(where x = 10, gets and inner cell):

```
*****|
12***|
x4***|
```

test: for loop to output neighbors array

expected result: neighbors array is

```
{', '1', '2', ', ', '4', ', ', ', '}
```

actual result:

```
{', '1', '2', '*', '4', ', ', ', '}
```

error fixed:

in the fourth assignment,

arr[x-1] changed to ' '

configuration 6: lr corner

(where x = 14, gets and inner cell):

```
*****|
***01|
***3x|
```

test: for loop to output neighbors array

expected result: neighbors array is

```
{'0', '1', ', ', '3', ', ', ', ', ''}
```

actual result:

```
{'0', '1', ', ', '3', ', ', ', ', ''}
```

configuration 7: inner cell

(where x = 7, gets an inner cell):

```
*012*|
*3x4*|
*567*|
```

test: for loop to output neighbors array

expected result: neighbors array is

```
{'0', '1', '2', '3', '4', '5', '6', '7'}
```

actual result:

```
{'5', '6', '7', '3', '4', ' ', ' ', ' '}
```

error fixed:

```
in the last three assignments,  
for (int y = 0; y < 3; y++); changed to  
for (int y = 5; y < 8; y++)...
```

file:test2.cpp, function: int countLive(...)

tested with gdb function calls

Many tests were done interactively throughout development,

but here is one example using gdb, showing a snippet from the command line:

ben@ubuntu:~/Public/cs162/project1\$ gdb test2

...

(gdb) b 44

Breakpoint 1 at 0x4009fc: file test2.cpp, line 44.

(gdb) run

Starting program: /home/ben/Public/cs162/project1/test2

warning: no loadable sections found in added symbol-file system-supplied DSO at 0x7ffff7ffa000

*****|

***|

X*****|

*****|

left edge

Breakpoint 1, main () at test2.cpp:45

45 std::cout << "around " << x << " are " <<

(gdb) p countLive(10, arr, width, len)

left edge

\$1 = 3

(gdb) p isLive(' ', 3)

\$2 = true

(gdb) p isLive(' ', 4)

\$3 = false

(gdb) quit

A debugging session is active.

Inferior 1 [process 4403] will be killed.

Quit anyway? (y or n) y

...

Tests after implementation:

This was simply running the final product and visually inspecting.