

Extra Tests for Assignment 4

Victor Nguyen

February 3, 2022

1 Description of pdf:

I just wanted to provide a few extra tests for students to use for their assignment 4.

2 Example test codes that you should check for:

2.1 Some sample codes that never terminates:

- `./life -s`
- `./life -s -t`
- `./life -s -t -n 10`
- `./life -o text.txt`
- `./life`

I think it's fine to allow these commands to not terminate, since the `./life` executable that the professor provides also doesn't terminate. Though, you should document somewhere that you know this is an issue and that you left it as is.

2.2 Other sample code that should be tested:

- `./life -i -t`

Your shell command should return something along the lines of: Error opening file.

3 Some input files you should test for:

I will provide a few files that you should probably test. Down below, I display the error that I get compared to what the example life executable displays. Whenever I type: `./life` this will refer to my `./life` executable, `./life_ex` will refer to the resource executable that was given.

3.1 init_neg_val.txt

The initial row/column contains a negative value. This text file should be formatted as:

```
3 -3
0 1
1 1
2 1
```

These are the results of the testing:

- ./life error - Segfaults
- ./life_ex error - Segfaults

Since the resources life executable returns a segfault, I will not be changing the return value of my own life executable. This should be noted though.

3.2 later_neg_val.txt

The row/column after the first line contains a negative value. This text file should be formatted as:

```
3 3
0 1
1 -1
2 1
```

These are the results of the testing:

- ./life error - Segfaults
- ./life_ex error - Malformed input

Since the resources executable returned a malformed input, I will also implement a way to return "Malformed input" when there is a negative number inside one of the row/column pairs.

3.3 mult_val.txt & mult_val2.txt

The file contains irregular formatting. This text file should be formatted as:

```
3 3 2
1 0
2
```

and

```
3 3 2
1 0
2 0
```

These are the results of the testing:

- ./life error - Segfaults/Weird display/Malformed input/Varies

- ./life_ex error - Varies

Depending on the irregularities in the input file, I sometimes get segfaults, sometimes the code runs the file but the ncurses displays something really weird, and sometimes I get segfaults. Because of this, I will implement my life executable to only accept the "traditional" formatting similar to the asgn4.pdf.

3.4 init_string_val.txt

The initial first line contains at least one string. This text file should be formatted as:

```
hello world
0 1
1 -1
2 1
```

These are the results of the testing:

- ./life error - Segfaults
- ./life_ex error - Malformed input

Since the resources executable returned a malformed input, I will also implement a way to return "Malformed input" when there are strings inside the first line.

3.5 later_string_val.txt

The row/column after the first line contains strings/characters. This text file should be formatted as:

```
3 3
0 1
hello -1
2 world
```

These are the results of the testing:

- ./life error - Segfaults
- ./life_ex error - Malformed input

Since the resources executable returned a malformed input, I will also implement a way to return "Malformed input" when there are strings after the first line.

3.6 out_of_bound.txt

The values after the first line are outside the universe. This text file should be formatted as:

```
3 3
0 1
5 1
2 1
```

These are the results of the testing:

- ./life error - Malformed input
- ./life_ex error - Malformed input

No changes will be necessary (at least on my end), this is because I've already implemented this during the process of creating universe.c.

4 Extra Notes:

- You should also make sure that these errors don't occur with **BOTH** stdin, and if the argument -i was passed through the executable.
- Testing and implementing the above problems **will not** guarantee a good grade on this assignment, it is just to provide extra testing for your code for a better grade. I could have missed a few more test cases.

5 Credit:

- Thanks to the discord user 190n#1979 aka ben for providing a few tests to look out for.