Middle East Technical University
Department of Computer Engineering

# CENG 111

Fall '19

Take Home Exam 3

## REGULATIONS

**Due date:** December 25, 2019, Sunday
*(Not subject to postpone)*

**Submission:** Electronically. You will be submitting your program source code written in a file which you will name as `the3.py` via CengClass. Resubmission is allowed (till the last moment of the due date). The last will replace the previous.

**Team:** There is **no** teaming up. The take home exam has to be done/turned in individually.

**Cheating:** This is an exam: all parts involved (source(s) and receiver(s)) get zero+parts will be subject to disciplinary action.

## PRELUDE

It is 2910... The year of the Ox... Or it would have been if humanity had survived. People realized too late to set aside their differences and instead focus on their real enemy. Well, they didn't know that there was an enemy.

In the last mission to exoplanet WASP-33b, astronauts brought some simple life forms that appeared harmless. Tiny little things they were. Humans called them froblins. They seemed to replicate a little and then die. Only a few survived for more than a handful of generations. But then some did. One form was in particular quite odd. It seemed to behave haphazardly for a long while until it turned into a feracious monster. It glided across the sky and wreaked havoc on humanity.

However, humans are not to be taken lightly. Despite the destruction caused by these unpredictable life forms, a small group of scientists managed to persevere in an underground research facility. They dedicated themselves to understand how these life forms behaved and reproduced. To their surprise, they found that their entire behaviour is governed by some very simple rules.

People thought they now have the upper hand. But not for long. Their overmind suspected that something was off. She sensed the presence of humans and their activities to undermine their dominance. So she changed the rules. The rules were no longer fixed. The only chance of the humanity was to learn what the new rules were and how it affects the spread of these creatures. Only then they could be prepared to fight what lurks in the darkness of the times to come.

## PROBLEM

You are among this small group of scientist who are working toward understanding the propagation of these froblins. Your team produced a model of their behaviour on a 2-dimensional grid. This grid is represented in a text file, which looks like:

```
- - - - -
- - - - -
- * * * -
- - - - -
- - - - -
```

The cells denoted by `-` are empty. Those denoted by `*` represent the froblins. Spaces are used to make the file more readable. Initially, your team discovered that their behaviour is governed by 3 simple rules:

```
- = 3 *
* < 2 -
* > 3 -
```

In each rule, the first character represents the cell type that the rule will affect. It can be `-` or `*`. The second character denotes an arithmetic relationship. It can be `=`, `<`, or `>`. The third character is a numeric value. It is a single digit. It represents how many froblin cells there are in the 8-neighborhood of a given cell. 8-neighborhood is defined as the four cells on the left, right, bottom, top as well as the four cells along the diagonals. The last character in each rule determines what to do in case the relationship is satisfied.

For example, the first rule says that: "All empty cells which have *exactly* three froblin neighbors, will become froblin cells in the next generation". The second rule says that: "All froblin cells which have *less than* two froblin neighbours will become empty in the next generation". Well, froblins don't like loneliness. Finally, the third rule says: "All froblin cells which have *larger than* three froblin neighbors will become empty in the next generation". Apparently, they also don't like to be overcrowded. It is important to note that the rules apply simultaneously to all cells in the map. This means that all updates must be made on a copy of the map in order to preserve the original state of the map as the rules are being applied.

## EXAMPLES

Below are some example evolutions of these scary froblin monsters. In these examples, "Generation 0" represents the input map.

### Example 1

```
Generation 0      Generation 1      Generation 2
 - - - - -         - - - - -         - - - - -
 - - - - -         - - * - -         - - - - -
 - * * * -         - - * - -         - * * * -
 - - - - -         - - * - -         - - - - -
 - - - - -         - - - - -         - - - - -
```

It then alternates between Generation 1 and 2 thereafter.

**Example 2**

Generation 0

```
- - - - - - - - -
- - - - - - - - -
- - - - - - - - -
- - - - - - - - -
- - - * * * - - -
- - - - * - - - -
- - - - - - - - -
- - - - - - - - -
- - - - - - - - -
```

Generation 1

```
- - - - - - - - -
- - - - - - - - -
- - - - - - - - -
- - - - * - - - -
- - - * * * - - -
- - - * * * - - -
- - - - - - - - -
- - - - - - - - -
- - - - - - - - -
```

Generation 2

```
- - - - - - - - -
- - - - - - - - -
- - - - - - - - -
- - - * * * - - -
- - - - - - - - -
- - - * - * - - -
- - - - * - - - -
- - - - - - - - -
- - - - - - - - -
```

Generation 3

```
- - - - - - - - -
- - - - - - - - -
- - - - * - - - -
- - - - * - - - -
- - - * - * - - -
- - - - * - - - -
- - - - * - - - -
- - - - - - - - -
- - - - - - - - -
```

Generation 4

```
- - - - - - - - -
- - - - - - - - -
- - - - - - - - -
- - - * * * - - -
- - - * - * - - -
- - - * * * - - -
- - - - - - - - -
- - - - - - - - -
- - - - - - - - -
```

Generation 5

```
- - - - - - - - -
- - - - - - - - -
- - - - * - - - -
- - - * - * - - -
- - * - - - * - -
- - - * - * - - -
- - - - * - - - -
- - - - - - - - -
- - - - - - - - -
```

Of course, if a different set of rules were used we would have observed different evolution patterns between generations. It is also possible that froblins are entirely eliminated after a certain number of generations, as shown by the next example.

**Example 3**

Generation 0

```
- - - -
- - * -
- * - -
- * - -
- - - -
```

Generation 1

```
- - - -
- - - -
- * * -
- - - -
- - - -
```

Generation 2

```
- - - -
- - - -
- - - -
- - - -
- - - -
```

All other generations will remain empty.

## OBJECTIVE

As the last hope of humanity, your goal is to write a Python program, called `the3.py`, that reads two input files that represent the initial map and the rules as well as an integer value that represents a generation count. Your program should output the map's state after simulating that many generations using the given rules. An example run will be as follows: `python the3.py map.txt rules.txt 56`. If `map.txt` contains the following data:

```
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - * * - - - - - - - - - - -
- - - - - - - - - - - * * - - - - - - - - - - - -
- - - - - - - - - - - - * - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
```

and the rules are as above, your program should produce the following output:

```
- - - - - - - - - - - - - * * * - - - - - - - - -
- - - - - - - - - - - * * - - - * - - - - - - - -
- - - - - - - - - - - * * - * - * - - - - - - - -
- - - - - - - - - - - * - - - * - * * - - - - - -
- - - - - - - - - - - - - - - - * * - * - - - - -
- - - - - - - - - - - * * * * - * - - * * * - - -
- - - - - - - - - - - - * * * * * - - - - - - - -
- - - - - - - - - - - - * - - * * - - - - - - - -
- - - - - - - - - - - - - * * - - - - - - - - - -
- - - - - - - - - - - - - * - - - - - - - - - - -
- - - - - - - * - - - - * - * - - - - - - - - - -
- - - - - - - * - - - - - * - - - - - - - - - - -
- - - - - - - * - - - - - * - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - * * * - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - -
```

## SPECIFICATIONS

- You are expected to implement a Python2 program that takes **3 command line arguments** in this order: the path to the map file, the path to the rules file, number of generations and **prints** the final state of the map.
- Your code should be run on inek machines with a command like this:
  `python the3.py map_path rules_path number_of_generations`.
- The map you print **should not** contain any other characters than "*", "-", and "\n". Do not print anything other than the final map.

- Map file contains a map with height and width being greater than or equal to 1.
- Rules file contains the rules line by line. Rules do not contradict with each other. There may be 1 or more rules in a file.
- number_of_generations is an integer greater than or equal to 0.
- In both files the lines are separated with “**\n**” characters.
- You can only use **sys** and **copy** modules. No other modules are allowed.
- You can use recursion or iteration, there is no limitation.
- Your codes will not be tested against erroneous inputs.

## GRADING

- Comply with the specifications. Since your returned results will be evaluated automatically, non-compliant results will be considered as incorrect by our evaluation system.
- Your program will be tested with multiple data (a distinct run for each data).
- Any program that performs only 30% and below will enter a glass-box test (eye inspection by the grader TA). The TA will judge an overall THE3 grade in the range of [0,30]. The glass-box test grade is not open to negotiation nor explanation.