Program Name: IR\_Minesweeper

Author: Jesmigel Cantos

Programming language used: Python

Prerequisites:

* Python interpreter
  + Compatible with both versions 2.\* and 3.\*
* Text editor
  + To be used for evaluating quality of work
* Command prompt for Windows, terminal for both Mac and Linux

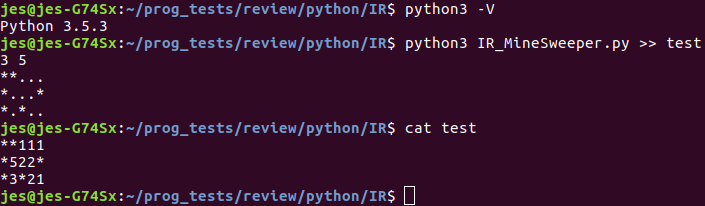
Usage Guidelines

* Assuming that a Python interpreter is implemented under a standard install, the program can be called upon as follows:
  + python IR\_Minesweeper.py – default python interpreter, usually python 2.\*
  + python3 IR\_ Minesweeper.py – forcefully invoke python 3.\* interpreter
* If python 2.\* is used, corresponding input should be enclosed within single quote ‘
  + The program is written with the intent of execution under a python 3.\* interpreter
  + Cause of the ‘ enclosure requirement involves the function input().
    - Under python 2.\* interpreters, input is processed under underlying eval() wherein input is treated as commands. ‘’ enclosure ensures that input will be treated as String.
    - Alternative would be the usage of raw\_input(). However, this function is not compatible with python 3
* If Python 3.\* interpreter is used, input is treated as string which would then can be casted into desired data types such as integers, floats or string.

Sample Screenshot:



**NOTE:** Output is undecorated and as per the above example, immediately displayed after the final input line. To capture large outputs, under a nix terminal >> may be used to redirect the standard output to an output file.



Code structure

* To show the level of comprehension regarding object oriented principles, an emulation of the Model-View-Control framework is used.
* In recent times, programming frameworks are commonly used due to its nature of being Agile.
* Implemented Classes are as follows:
  + Model Class: model\_MineField
    - Define the structure of a minefield object
  + Controller Class: controller\_OperateData
    - This class is responsible for interrogating the standard input to promote quality assurance.
    - After passing quality assurance, stored data will then be interrogated and analysed in order to achieve desired output as stated by the assignment requirements
  + View Class: view\_initializeObjects
    - Normally the "Front end" for user interaction. With respect to the task at hand, This will call objects meant for input, storage, and processing of data
* Implemented “High level Function” are as follows
  + Function: Main
    - Responsible for calling different type of "views"
    - This acts as the singleton design pattern where in theory, other design views are going to be called upon based on the input. This can be combined with other design patterns such as either of the following:
      * factory, abstract factory, and command handler
  + Function: testClass
    - This will act as the "Test class" where if required test methodologies can be implemented.
    - In theory, the minesweeper field can be analysed by means of regular expressions. If data is joined into a single string, 4 regex lines can be made for look ahead and look behind logic
  + Function: call
    - Immediately invoke functions either Main(), testClass() or both
* Appropriate input error handling has been applied as follows
  + Try-except ladder is implemented to invoke clean exit with appropriate message if recognised error is identified
    - except(*error parameters*) as *error\_variable*
      * sys.exit(“*error message*” + str(*error\_variable*))

**NOTE:**

* As per the nature of this assignment, same result can be produced with the usage of a single loop and combination of 4 regular expression statements for look-ahead, look-behind, string replace, and perform operation logic.
* However, as per the assignment requirements, this method is avoided in order to show object-oriented concept comprehension