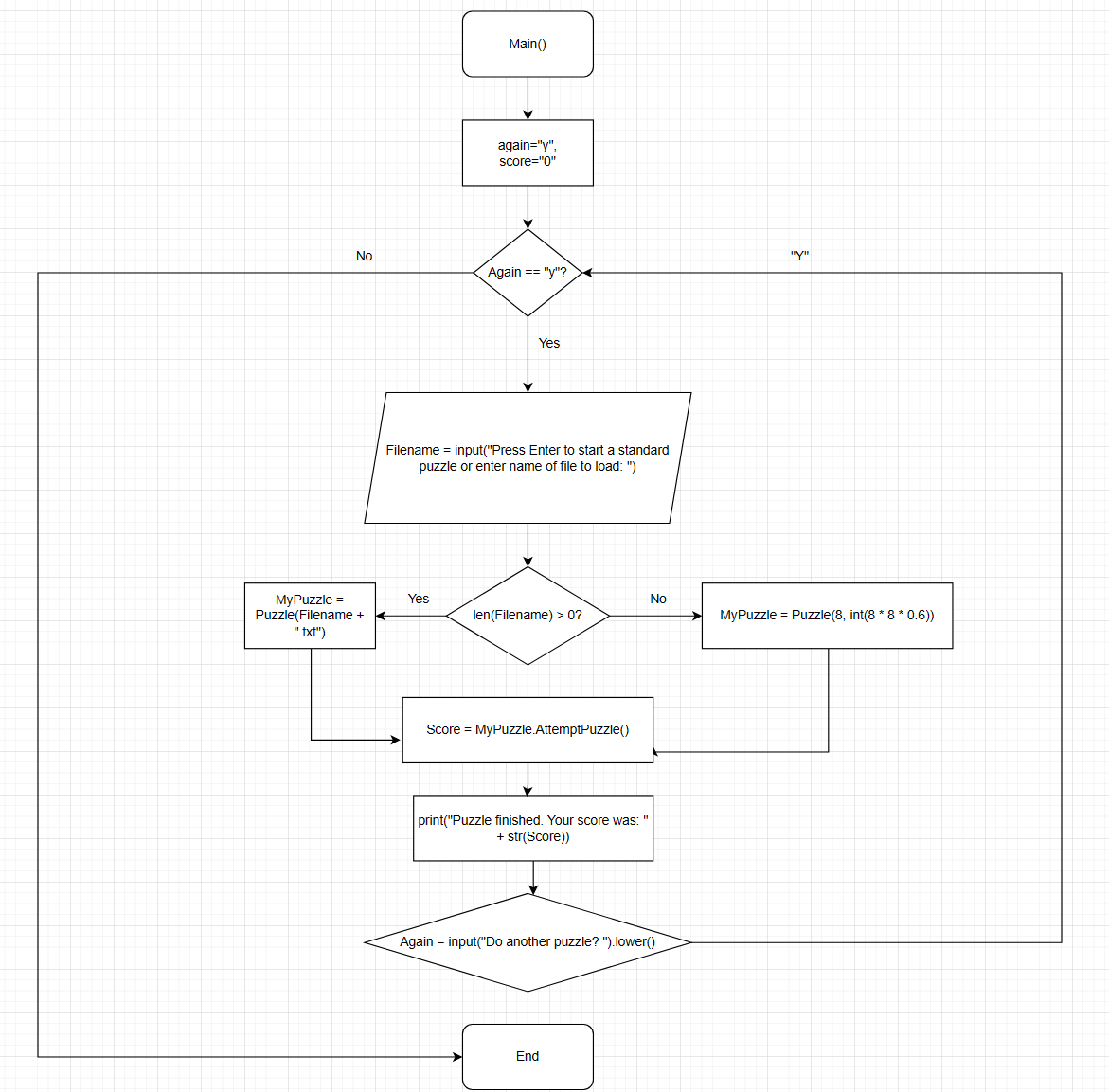
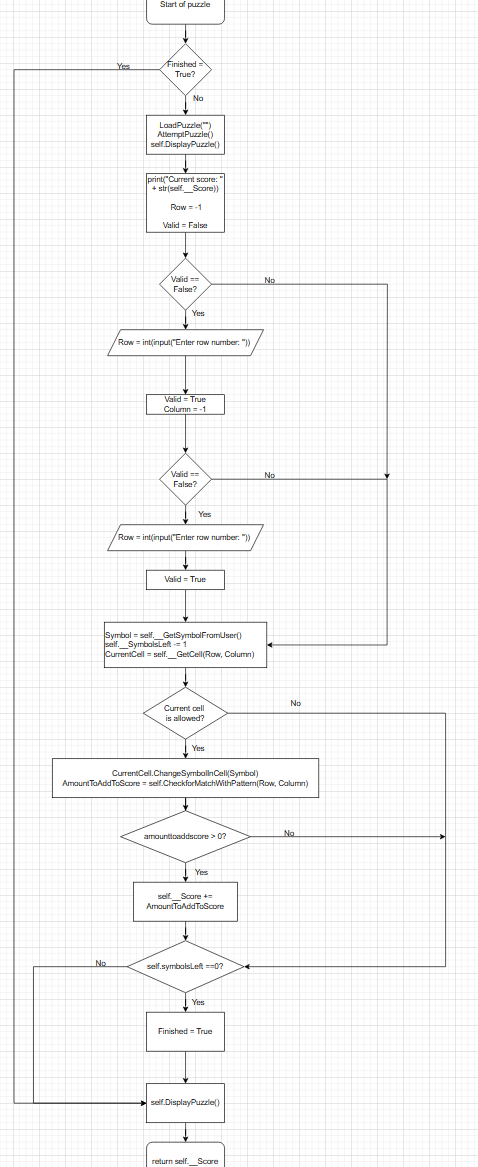
**Skeleton Program Questions**

1) Flowcharts for procedures:

Main() function flowchart:

****

puzzle flowchart:

****

2) How the program checks if a pattern has been completed:



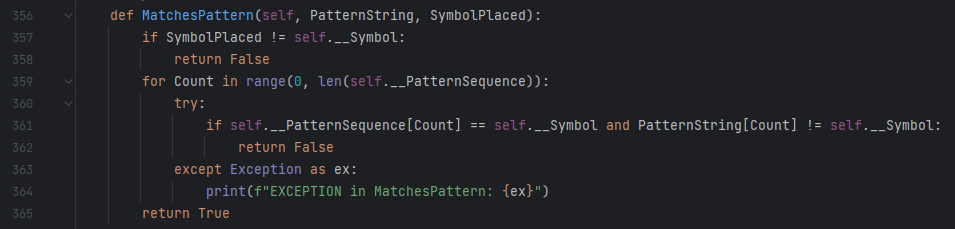
The **CheckforMatchWithPattern()** subroutine loops through a 3x3 section of the grid, starting from the specified row and column.

-“PatternString” is created from the symbols in the cells in the region.

-This constructs a 3x3 grid with symbols in it, the string will look something like “X\*\*XXX\*\*X”, and symbols are read/constructed in a clockwise manner.

-The subroutine then starts to iterate through all the allowed patterns, and sees if the subroutine **MatchesPatern()** returns true

-If it returns true, then a score of 10 is returned, if not, then a score of 0 is returned.



The MatchesPatern() subroutine will take a pattern string as a parameter, and what the pattern symbol is.

-First, if the symbol placed on the grid is not the same as the symbol of the specific pattern, False will be returned, as a pattern would have not been found

-If it’s the same symbol, then a for loop starts, comparing each position in the “PatternSequence” to corresponding position in the “PatternString”, to check if they match.

-If any position is found not to match, then False will be returned

-If the loop finishes without finding any positions that do not match, then True will be returned, as the patternSequence would have been fully matched with the PatterString

3) bugs/unfinished parts:

-CheckSymbolAllowed only returns False (does nothing)

-Blocked cell class does nothing, init only supers form blockedCell and has one symbol parameter

-UpdateCell subroutine in Cell class does nothing (just passes)

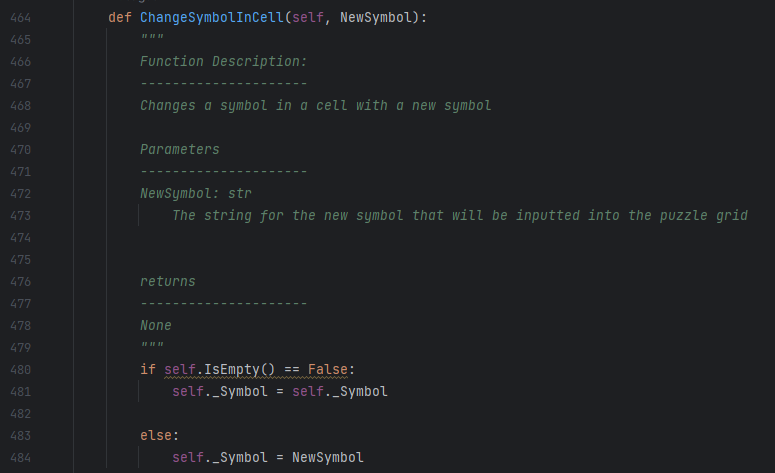
-cells can be overwritten in the grid

-score for puzzle is unfinished for puzzles

-if a filename that doesn’t exist is inputted, you are put in an infinite loop

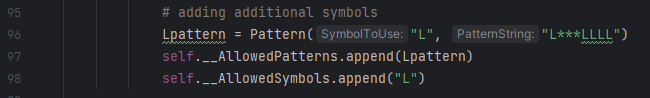
4) Blocking a cell once it has been entered:

Tweaked ChangeSymbolInCell() subroutine in the Cell() class, so that if the cell is not empty (its already been entered to), then the cell can’t be changed again (blocked)



5) adding a new symbol pattern for “L”:

In the Puzzle() class \_\_init\_\_() function:



7) Likely extensions they may ask:

-fix game file not existing

-ability to remove a blocked cell at the cost of score

-make symbol case not matter

-add difficulty

-save the current game

-undo a move

-create a new puzzle file

-check for row and column validation so that they stay in the grid