At home research notes:

rstring() deletes whatever characters are placed into a the () defaults to blank space and this will erase anything to the right (after) of the string.

Example:

>>> ' spacious '**.**rstrip()

' spacious'

>>> "AABAA"**.**rstrip("A")

'AAB'

>>> "ABBA"**.**rstrip("AB") *# both AB and BA are stripped*

''

>>> "ABCABBA"**.**rstrip("AB")

'ABC'

lstring() deletes whatever characters are placed into the () defaults to blank space and this will erase anything to the left (before) of the string.

Example:

>>> ' spacious '**.**lstrip()

'spacious '

>>> "AABAA"**.**lstrip("A")

'BAA'

>>> "ABBA"**.**lstrip("AB") *# both AB and BA are stripped*

''

>>> "ABCABBA"**.**rstrip("AB")

'ABC'

>>> "ABCABBA"**.**lstrip("AB")

'CABBA'

string() deletes a specific character in a string

Example:

>>> ' spacious '**.**strip()

'spacious'

>>> "AABAA"**.**strip("A")

'B'

>>> "ABBA"**.**strip("AB")

''

>>> "ABCABBA"**.**strip("AB")

'C'

Rjust() adds to front of string spaces or characters denoted in ()

Example:

>>> "ABC"**.**rjust(10)

' ABC'

>>> "ABC"**.**rjust(10, "#")

'#######ABC'

>>> "ABC"**.**rjust(2, "#") uses the 2 when counting spaces bc abc is 3 spaces then no # are added

'ABC'

Ljust() adds to back of string spaces or characters denoted in ()

Example:

>>> "ABC"**.**ljust(10)

'ABC '

>>> "ABC"**.**ljust(10, "#")

'ABC#######'

>>> "ABC"**.**ljust(2, "#”) uses the 2 when counting spaces bc abc is 3 spaces then no # are added

'ABC'

Center() centers the string in between the denoted character or spaces

>>> "ABC"**.**center(10)

' ABC '

>>> "ABC"**.**center(10, "#")

'###ABC####'

>>> "ABC"**.**center(2, "#")

'ABC'

while loop to check for blank lines:

???