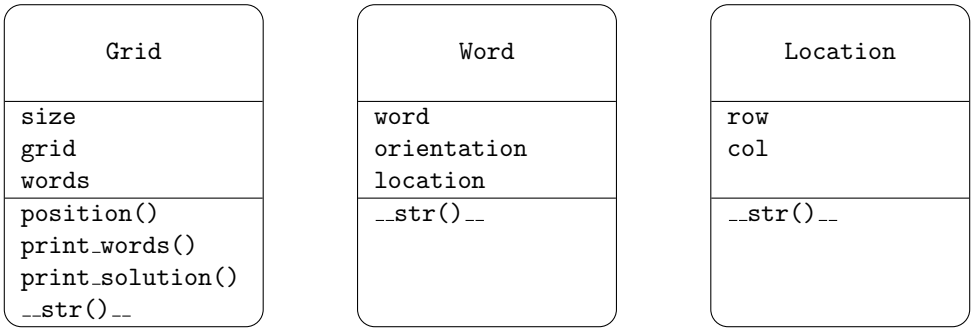
In the last programming assignment, you implemented the Word and Location classes. A test program (WordSearch.py) was used to test your implementation of the two classes. In this programming assignment (which is part 2), you will focus on the Grid class and the main program (we will call this one WordSearch.py again) that drives the process of creating the word search.

```
U C Y Z G A D D P E
W M E K A N S Z N L
K Q Q J Y L Y P W I
L Z S A A X A E O D
I A E M E N Q W F O
P D F B T A U K S C
G Q U H R K S B K O
U C E I H A S O S R
C R D I G J N P X C
H A L L I G A T O R
```

Specification

You will only focus on the `Grid` class and use the `Word` and `Location` classes from the first part.



For this assignment, the following are provided to you:

- animals.txt
- words.txt
- WordSearch.py: the main part of the program that makes use of the Location and Word classes.
- Location.py - This is the class that you have generated in Part I
- Word.py - This is the class that you have generated in Part I
  - In the word class, define the __lt__(self, other) subroutine, which will be useful to sort words.

Grid Class

- The given Grid.py is partially filled out for you.
- Constructor
  - The default values of `size` in the constructor should be 25
  - Declare the variable `grid` as an empty list.
  - Declare the variable `words` as an empty list

  - Initalize the `grid` by adding blank rows to the list. Each row is another list of Grid.BLANK equal to the `size`.

- Define getters for size, grid and words. Appropriate setters for size. If negative values are specified through the setters, the values should be reset to 25.
- In the Grid class, you must add support for the remaining orientations (other than the provided HR orientation). This includes appropriately setting `min_row, max_row, min_col,` and `max_col` in the `position()` subroutine
- Appropriately set the `row` and `col` in the `_check()` function; and `row` and `col` in the `_position()` subroutine.
- In the Grid class, you must sort the list of words prior to displaying them in the print_words() function using the list.sort() function.
- Appropriate comments are added in the Grid.py identifying the missing code parts.

Sample Outputs

Make sure all the related files/classes are in the same folder. Note that the output varies on every run, as the words are randomly chosen. For the number of words as 15 and grid size of 25, running WordSearch.py with animals.txt has the folllowing output:

```
Successfully placed 15 of 15 words.

O V U N B S R U N F I J O Y G V I E H C T N Q M S
X L Y T T T O I L R Y L X O J X J O K U H C P L W
V S L R M T A P I J U F Y A H S X F P H I D R V Z
M N T I I T U U L M T B D S C U H C I A N A U G I
V G J Q D T G A A D P E N G U I N J F C I V G N Y
X N Y L E A Z Q F N D C O V M C V F Y D D N W E F
S E Z D H W M J N Q R E O A L B A T R O S S R C Y
Y L F Q N Q G R A F V H R Q L F E B U F B S X V G
N I J Y E P Q M A F Y Y A O O X J K D B V N A Z I
U D I D R T G I E E E F G O P L R R Y N N G G N Z
N O H U G A U F N N U N N F H Q P S U R N S O N I
C C T W Q N V A R Z L W A X L X G T H L C B R V J
P O Q O A M A V R D C Z K X E Y I B E L T X F X X
K R L B R T U R K E Y H J M L T J A I P Z T O I X
B C H N K R L N D B R P X W G L D F I O N M U C D
H U W M E J A B J E H U Z U A O I I B Y N N N A C
I O R M R E E P H U N T Y E E N N T M F O O L J P
M S N V G A L T K S V P D O K J E C Q U P L X K A
S K K J S L N S G B U O I B U H C M I E I B T V N
P C R U T A R N H G X E O T M M W E C G M S O P K
D Z S X P V F J C A N A M Z H S M Q A T B R P J D
F E O W A W S I O O R N J H A J O T G T C I X H A
E L T R U T N K G S W K K O S H O L I K H M O O N
T O L R Z Y E Y C P O C S P E R C P X K C L N L J
P T N B P Y E C C M B F M Y C G F D W L X B J A J

ALBATROSS/HR@(6,13)
ALLIGATOR/DLD@(15,23)
ARMADILLO/DLU@(8,8)
CROCODILE/VU@(14,1)
EAGLE/VU@(16,14)
```

```
FROG/VU@(12,22)
HYENA/DLD@(7,11)
IGUANA/HL@(3,24)
KANGAROO/VU@(12,12)
PANTHER/DRU@(20,4)
PARROT/DLU@(16,7)
PENGUIN/HR@(4,10)
SHARK/DRD@(18,7)
TURKEY/HR@(13,5)
TURTLE/HL@(22,5)
```

```
O . . . . . . . . . . . . . . . . . . . . . . .
. L . . . . . . . . . . . . . . . . . . . . . .
. . L . . . . . . . . . . . . . . . . . . . . .
. . . I . . . . . . . . . . . . . . . A N A U G I
. . . . D . . . . . P E N G U I N . . . . . . . .
. . . . . A . . . . . O . . . . . . . . . . . .
. E . . . . M . . . . . O A L B A T R O S S . . .
. L . . . . . R . . . H R . . . . . . . . . . . .
. I . . . . . A . Y . A . . . . . . . . . . . . .
. D . . . . . E . . G . . . . . . . . G . . .
. O . . . . . N . . . N . . . . . . . . O . . .
. C T . . . . A . . . . A . . . . . . . . R . . .
. O . O . . . . . K . E . . . . . . . F . . .
. R . . R T U R K E Y . . . L . . . . . . . . .
. C . . . R . . . . R . . . G . . . . . . . . .
. . . . . A . . E . . . . A . . . . . . . . A .
. . . . . P H . . . . . E . . . . . . . L . . .
. . . . . T . . . . . . . . . . . . . . L . . . .
. . . . . N S . . . . . . . . . . . . . I . . . .
. . . . . A . . H . . . . . . . . . . . G . . . . .
. . . . P . . . . A . . . . . . . . . A . . . . . .
. . . . . R . . . . . . . T . . . . . . . .
E L T R U T . . . . . K . . . . O . . . . . . . .
. . . . . R . . . . . .
. . . . . .
```

Deliverable

- Submit the Word.py, Location.py and Grid.py

Rubric

Item	Points
Good coding style	2
Appropriate Comments & Header	4
Grid class	3
Grid class constructor	3
Grid class getters	4
Grid class setters	4
Grid position function updated	4
Grid check function updated	4
__lt__ function imlemented in Word class	4
Output is correct	8
Total	40

FIN