

```

function (create word search):
open words.txt file
for each word in words.txt:
    if the current word is less than 4 letters in length:
        remove the current word from words.txt
    else:
        update the current word to be in lower case in words.txt
set x and y number to 0
set y coordinate to 0
while y coordinate is less than 11:
    set x coordinate to 0
    while x coordinate is less than 11:
        if the x and y coordinates are not equal to 0:
            add a random letter to the grid from a frequency representative table
            at the coordinates x, y
        elif the y coordinate is not equal to 0:
            add the x number in the grid to the coordinate x, y
            increase this x number by 1
        elif the x coordinate is not equal to 0:
            add the y number in the grid to the coordinate x, y
            increase this y number by 1
        increase x coordinate by 1
    increase y coordinate by 1
while there is less than or equal to 10 words in the answers:
    choose a random x and y coordinate between 0 and 10 as the start coordinates
    choose a random direction, a number between 0 and 7
    choose a random word from words.txt
    check if collisions will occur and place the word with the function (check word)
    update the grid using the temporary grid
    choose a random number from 0 to 1
    if the random number is greater than or equal to 0.1 and a collision did not occur:
        add the word to the answers with a start and end coordinate
        remove the random word from words.txt
add the answers to words.txt
close words.txt file
return the grid and the answers

```

```

function (check word)
make a temporary grid by copying the grid
find were next coordinates will go with the function (next coordinate)
while letters added to the temporary grid are less than the length of the word:
    if collision with another word has not occurred or the letters at the current coordinates
    are the same:
        add the current letter to the temporary grid at the current coordinate
        move the current coordinate to the next
    else:
        return the original grid
return the temporary grid and the last coordinate

```

```
function (next coordinate)
if direction is 0:
    the next coordinate will be one up
elif direction is 1:
    the next coordinate will be one up and one right
elif direction is 2:
    the next coordinate will be one right
elif direction is 3:
    the next coordinate will be one down and one right
elif direction is 4:
    the next coordinate will be one down
elif direction is 5:
    the next coordinate will be one down and one left
elif direction is 6:
    the next coordinate will be one left
elif direction is 7:
    the next coordinate will be one up and one left
```

```
function (check answer):
open words.txt
ask the user to input the word that is being guessed
check if a keyword has been inputted with function (keywords)
if input is not in words.txt:
    display error
    rerun the function (check answer) due to an error occurring
display that the word is valid
close words.txt
while not all the coordinates have been entered:
    ask the user to input the current coordinate
    check if a keyword has been inputted with function (keywords)
    if input is an integer and is between 1 and 10:
        display valid coordinate
        store input as the current coordinate
        change current to the next coordinate
    else:
        display invalid coordinate
combine the start x and start y coordinates as the inputted start coordinates
combine the end x and end y coordinates as the inputted end coordinates
if the inputted start coordinates or inputted end coordinates is not equal to the coordinates of
the inputted word:
    display error
    rerun the function (check answer) due to an error occurring
return the word correctly found
```

```

function (keywords)
if input is the word FINISH:
    display FINISH
    FINISH
elif input is the word RESTART:
    display RESTART
    RESTART
elif input is the word ANSWERS:
    display ANSWERS
    if no total time has been recorded yet:
        record the total time taken with function (timer)
    display the word search board with all the words correctly guess with the function
    (display board)
    calculate the current points with the function (calculate points)
    display the time, points and words correctly found
    RESTART

```

```

function (display board)
display the word search grid
for each word in the answers:
    display the word next to this grid but below the previous word
display a rectangle around the grid and the answers with no fill and a black outline behind
the grid and the answers
display a rectangle around the answers with a light blue fill and a black outlined behind the
grid and the answers
for each word correctly found:
    display a rectangle with curved ends from the start coordinates to the end coordinate
    with a red outline and no fill.

```

```

function (calculate points)
while there are unprocessed found words and time taken to find the word is less than 5
minutes:
    add the word length to the points
return points

```

```

function (timer)
if time is zero:
    record the staring time
else:
    record the ending time
    subtract starting time from ending time
return the time

```

START

display welcome message

display instructions

display what the keywords do

generate the word search with function (word search)

record the starting time with function (timer)

display the word search board with the function (display board)

while not all the words have been correctly found:

 record the current time taken with function (timer)

 calculate the current points with the function (calculate points)

 display points scored

 display the word search board with the function (display board)

 run the function (check answer)

record the total time taken with function (timer)

display word search is completed

display the word search board with the function (display board)

display points, time taken, and words correctly found

while no keywords have been inputted:

 ask the user if they want to type FINISH to finish, AGAIN to restart or ANSWERS to

 display the answers

 check if a keyword has been entered with function (keywords)

FINISH