ECE 101: Programing I
Final Project: *Word Finder*
Due Date: Monday December 8, 2025 at 4.00 pm, on zyBooks

All due date:

- Individual/Group sign up - Wednesday November 19, 2025 by 11.59 pm
- Final project - Monday December 8, 2025 at 4.00 pm

1. Administrative details:

1.1 Group: You can work on this final project ***by yourself***, or you can work in a ***group of (maximum) 2*** students. If you choose to work as a group, use collaborative tools to facilitate the project development, such as replit.com, google doc, etc.

Sign up your team or if you will work by yourself for the final project https://forms.gle/uKowdhBYFa7PAphE7

The team/group must be signed up by 11.59 pm Wednesday November 19, 2025. No new group can be formed after this date (unless approved by an instructor). You can check that your group signed up properly at https://docs.google.com/spreadsheets/d/1lzlNeaqo4v96L3hjAm4CtwkUKNTzGVZ__7WBrGLnt44/edit?usp=sharing

1.2 Final Code Submission: The project is due on **Monday December 8, 2025 at 4.00 pm. Note that the due time is 4 PM.**

If you work as a team, only one team member must submit the code. Make sure that both your names are on the top of the code as comments.

The same score will be given for the team whose names are in the sign-up sheet in section 1.1. If both your names are not on the team sign up sheet, you are "not" a team.

Late submission policy: 20% deduction per day.

The ***last day*** of final project submission is by 4.00 PM on Wednesday December 10, 2025. After this date/time, the final project will NOT be accepted/graded.

1.3 Academic Integrity Policy: Each team is expected to submit its own code. You may ask others for advice and/or discuss the project in general. However, you/your group MUST **WRITE YOUR OWN CODE.**

If any part of the code submitted by other students or other teams is identical, ALL involved parties will receive **0** credit on the entire project and one letter reduction in their final grade. This policy will be very aggressively enforced. ALL submitted code will be checked with a plagiarism detection tool on zyBooks.

1.4 Points distribution: See Grading Rubrics (page 5 and 6) for the final project grading.

1.5 Suggestions:

a) Spend time designing your code and use modular programming. Create all function prototypes and describe what they do before developing your code. Create pseudocode of your program.

b) Determine test cases for your functions and your overall code to ensure proper functionality.

c) Write well-documented code.

2. Project description – Word Finder

Word finder (or word scramble) is a word game where you are given a set of letters and you rearrange the letters to create original words. For example, given

i a n m c a

original 6-letter words that can be created are maniac, caiman

original 5-letter words are amain, amnia, amnic, anima, mania, manic

original 4-letter words are acai, amia, amin, cain, cami, main, mana, mica, mina

There are also original 3-letter words and 2-letter words that can be generated. However, in our game, we will use words starting from 4 letters and up.

Project requirements: Write an **interactive C program** for Word Finder with the following requirements:

2.1 Text files:

When developing the code, use one text file, WordSet2.txt.

After your code is working correctly for the above text file, use a rand ( ) function to generate a number between 0 – 9 in order to pick a text file to use. There are 10 text files, called WordSet0, WordSet1, …, WordSet9, given for your Word Finder game. Make sure that your program works for all 10 text files.

2.2 Array of struct:

Use the words given in the text file(s) to create an array of the following struct. **Your code must use the following struct.**

```
typedef struct wordnode_s {
        char word[15];
        int point;          //points assigned for each word
} wordnode;
```

Note: You can add more members or you can create additional struct data type.

2.3 Scrambled letters: Display a scrambled version of letters that can be used.

2.4 Clues display: Display *at most 4 clues* to help the players. All vowels should be displayed in each clue. Display the 4 clues using the first two words and the last two words in the array (given that those words have not been correctly guesses). If fewer than 4 clues are left, display all clues.

2.5 Let a user guess one word: Since our game starts with 4 letters and up, your program will continue asking if the player enter a word that has fewer than 4 letters or has more than the max number of letters in the longest word in the text file (observe that the first word in every text file is the longest word).

2.6 Decide whether the entered word is correct or not.

- If the entered word is one of the words in the text file (the array created in 2.2), add more points to the total points that the player gets. Display a statement that the player guesses one word correctly and the total points.
- If the entered word is an incorrect guess, deduct number of guesses that the player has (maximum number of guesses that a player has is 10). Display a statement that the player guesses wrong and number of guesses that the player has left.

2.7 Display correctly guessed words and wrongly guessed words

2.8 Game ending: The game can be ended in 3 following ways:

a) After every turn/guess, your code asks whether the player wants to continue (q to quit).
   If the player quits, your program displays all unguessed words.
   If the player continues, repeat step 2.3 – 2.8.
b) The player guesses all words (in the array/text file) correctly.
c) The player uses up all 10 guesses (maximum number of guesses that a player has is 10).

Optional Features for Extra Credit (EC)

Note: Each optional feature must be correctly and fully functional in your code in order to get the extra credits for that part. **No partial extra credit will be given for somewhat functioning feature(s).**

- EC1 (4 points) Graphic: use graphics to display scrambled letters. See item 4) in the List of Sample code execution below.

- EC2 (4 points) Add features (user input prevention): When the player guesses a word, your program keeps asking
  a) if the player enters a word containing letters that are not used in the game. For example, given i a n m c a, if the player enters "name", your program will ask for the next word/guess since e is not in the set of letters which are i a n m c a.
  b) if the player guesses the same word (correct or wrong), your program will let the player know and ask for the next word/guess. For example, assuming that,
     correct guesses: main
     wrong guesses: acni
  Then the player enters acni (again), your program will let the player know that the player already entered this word and ask the player to enter another word/guess.

List of Sample code execution:

1) Sample 1 (using WordSet2.txt) – when the players do not enter valid words, got a few correct and wrong guesses and then quit.
2) Sample 2 (using WordSet2.txt) – when the players used up all 10 guesses.
3) Sample 3 (using WordSet2.txt) – when the players won (correctly guess all words in the text file).
4) Sample 4 - show graphics (extra credit) – to get full score for these extra credits, all letters in a word must be displayed on the same line. If you have different ideas for the graphics, please talk to your instructor to get approval before implementing it.
5) Sample 5 (using WordSet1.txt) – playing the game and then quit.

Final project grading rubric

	Category	Maximum points	Excellent (100%)	Good (70%)	Marginal (20%)	Unacceptable (0%)
	Text file	4	Successfully use rand( ) to randomly choose the text file to be opened to read. Use fopen and fclose correctly	No rand( ) used – hardcode the text file name. Use fopen and fclose correctly	Some code skeleton but the code is not working	No implementation
	Populate the array of struct with the text file	14	The array is populated with every word in the chosen text file.	Not all words in the text file are present in the array	Some code skeleton that attempts to do this part but unsuccessful	No implementation
	Scrambled version of letters	6	The letters can be properly scrambled	The letters are somewhat scrambled (rand function does not use correctly)	Some code skeleton that attempts to do this part but unsuccessful	No implementation
	Display at most 4 clues (first 2 and last 2 words of the array)	8	• First two and last two words are displayed. • The vowels in each word are displayed. • If the word in the clue is removed due to correct guess, the next word in the array can be properly used as a clue. • Properly display when fewer than 4 clues left	Some functionalities are modified such as • using the first 4 words instead (-2 pts), • clues displayed but no vowel (or first letter is displayed instead) (-2 pts), • not properly working (or the code breaks) if fewer than 4 clues are left (-3 pts)	Some code skeleton that attempts to do this part but unsuccessful	No implementation
	Let a user enter valid words	4	The code lets a user enter a word and continues asking if the word length is not between 4 and the length of the longest word in the array	The code lets a user enter a word with any length (-2 pts)	Some code skeleton that attempts to do this part but unsuccessful	No implementation

	Decide whether the word is correct or not	4	The code can correctly decide whether the word is a correct guess.		Some code skeleton that attempts to do this part but unsuccessful	No implementation
	Handle a correct guess	5	• If correct, the total points are calculated correctly • The statement(s) are displayed correctly	There are some miscalculations, or the code cannot get the point from each array element (-2 pts)	Some code skeleton that attempts to do this part but unsuccessful	No implementation
	Handle a wrong guess	5	• If wrong, the number of guesses is deducted correctly. • The statement(s) are displayed correctly		Some code skeleton that attempts to do this part but unsuccessful	No implementation
	Display correctly guessed words and incorrectly guessed words	12	The code can correctly display the correct-guessed words and the wrong-guessed words	Some functionalities are missing or not working such as only correct or wrong guesses is displayed (-5 pts)	Some code skeleton that attempts to do this part but unsuccessful	No implementation
	Correct words handling	10	The correctly guessed words are handled correctly. They cannot be used as clues again.		Some code skeleton that attempts to do this part but unsuccessful	No implementation
	Ask whether a user wants to continue (q/Q to quit)	4	The code successfully lets a user enter a letter and can quit or continue		Some code skeleton that attempts to do this part but unsuccessful	No implementation
	When quit, print number of words and words that have not been guessed	4	All unguessed words are correctly displayed.	Some functionalities are missing or not working such as not all unguessed words are printed.	Some code skeleton that attempts to do this part but unsuccessful	No implementation
	If all words are guessed correctly, congrats and total points	4	The code can successfully end the game when a player correctly guesses all words and print the correct total points	Some functionalities are missing or not working such as the game does not end even though all words are correctly guessed or wrong total points (-2 pts)	Some code skeleton that attempts to do this part but unsuccessful	No implementation

	If the player uses up all 10 guesses	4	The code can successfully end the game when a player uses up all 10 guesses		Some code skeleton that attempts to do this part but unsuccessful	No implementation
	Program Design					
	Code modularity	4	The code is logically divided into functions that implement important functionality	The code is modular but further simplification could have been done		All statements are written in main function
	Code documentation	2	Comments are used throughout the code. The input/output and goal of every function is adequately described.	The code is partially documented	The code is scarcely documented	No comment written in the code
	Compilation	6	Code successfully compiles without syntax errors. The code does not hang while in execution	The code successfully compiles, but some conditions make it hang while running (-3 pts)		The code has syntax errors at the compilation time (-6 pts)
	Total	**100**				
	Extra credits		Excellent (100%) – No partial credit (all or nothing)			
	Graphic	4	See Extra credit section of this handout for details			
	User inputs prevention	4	See Extra credit section of this handout for details			

Sample code execution 1 (using WordSet2.txt): **Bold** information entered by a user.

```
|-----------------------------------|
|-  LET's Play Word Finder  -|
|-----------------------------------|

@@@ --- @@@ Letters in words @@@ --- @@@
      p i t d n u
@@@ --- @@@ --- --- --- --- @@@ --- @@@

You have 10 guesses left
Clues to help you guess
u _ i _
_ u _ _
_ u _ _ i _
i _ _ u _

Enter your guess (4-6 characters): abc
Enter your guess (4-6 characters): abcdefgh
Enter your guess (4-6 characters): pit
Enter your guess (4-6 characters): pundit

$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$
Yay ... you got 50 points
Total points: 50
$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$

Correct guesses: pundit
Wrong guesses:

continue (q to quit): g

@@@ --- @@@ Letters in words @@@ --- @@@
      u n i d p t
@@@ --- @@@ --- --- --- --- @@@ --- @@@

You have 10 guesses left
Clues to help you guess
u _ i _
_ u _ _
i _ _ u _
_ i _ _

Enter your guess (4-6 characters): unit

$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$
Yay ... you got 10 points
Total points: 60
$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$

Correct guesses: pundit unit
Wrong guesses:
```

Words and points in WordSet2.txt
pundit 50
input 30
dint 10
pint 10
punt 10
unit 10

The letters of the first word (longest word) are scrambled and shown here

Start with 10 guesses

The first 2 clues are from unit and punt and the next 2 clues are from pundit and input. Only vowels are shown.

keep asking the players to enter a guess until they enter a word between the required length.

pundit is a correct guess, and the players now get 50 points (50 is the points given for pundit in the text file)

Correct and wrong guessed words are displayed

Enter any letter by q to continue

The letters of the first word are scrambled again (observe that they are not in the same order as the above)

Still 10 guesses left since the players guessed the correct word in the last turn

The first 2 clues are still from unit and punt and the next 2 clues are from input and dint (since pundit is already correctly guessed and it should not be a clue.

unit is a correct guess, and the players now get 60 points (10 more points from unit)

Correct and wrong guessed words are displayed

continue (q to quit): **j**
@@@ --- @@@ Letters in words @@@ --- @@@
t d i n u p
@@@ --- @@@ --- --- --- --- @@@ --- @@@

You have 10 guesses left
Clues to help you guess
_ u _ _
_ i _ _
_ i _ _
i _ _ u _

Observe that the clues are changed since unit and pundit are already correctly guessed.

Enter your guess (4-6 characters): **testers**
Enter your guess (4-6 characters): **put**
Enter your guess (4-6 characters): **dunt**

:(--- :(--- :(--- :(--- :(--- :(--- :(
Oops ... you lost 1 guess
You have 9 guesses left
:(--- :(--- :(--- :(--- :(--- :(--- :(

Wrong guess -> lost 1 guess and the wrong word is now on the “wrong guesses” list.

Correct guesses: pundit unit
Wrong guesses: dunt

continue (q to quit): **j**
@@@ --- @@@ Letters in words @@@ --- @@@
d u i t p n
@@@ --- @@@ --- --- --- --- @@@ --- @@@

You have 9 guesses left
Clues to help you guess
_ u _ _
_ i _ _
_ i _ _
i _ _ u _

Enter your guess (4-6 characters): **dupt**

:(--- :(--- :(--- :(--- :(--- :(--- :(
Oops ... you lost 1 guess
You have 8 guesses left
:(--- :(--- :(--- :(--- :(--- :(--- :(

Wrong guess -> lost 1 guess and the wrong word is now on the “wrong guesses” list.

Correct guesses: pundit unit
Wrong guesses: dunt dupt

continue (q to quit): **q**

|--- G o o d --- T r y ---|
|--- You got 60 points ---|

The players quit. Display the total points and the unguessed words.

Unguessed words:
input, dint, pint, punt,
Good Game ... Good Bye

Sample code execution 2 (using WordSet2.txt): **Bold** information entered by a user.

```
|-----------------------------------|
|-  LET's Play Word Finder  -|
|-----------------------------------|

@@@ --- @@@ Letters in words @@@ --- @@@
      n  u  i  t  p  d
@@@ --- @@@ --- --- --- --- @@@ --- @@@

You have 10 guesses left
Clues to help you guess
u _ i _
_ u _ _
_ u _ _ i _
i _ _ u _

Enter your guess (4-6 characters): pit
Enter your guess (4-6 characters): dipt

:( --- :(  --- :( --- :( --- :( --- :( --- :(
Oops ... you lost 1 guess
You have 9 guesses left
:( --- :(  --- :( --- :( --- :( --- :( --- :(

Correct guesses:
Wrong guesses: dipt

continue (q to quit): g
@@@ --- @@@ Letters in words @@@ --- @@@
      n  p  i  d  u  t
@@@ --- @@@ --- --- --- --- @@@ --- @@@

You have 9 guesses left
Clues to help you guess
u _ i _
_ u _ _
_ u _ _ i _
i _ _ u _

Enter your guess (4-6 characters): dipt
:( --- :(  --- :( --- :( --- :( --- :( --- :(
Oops ... you lost 1 guess
You have 8 guesses left
:( --- :(  --- :( --- :( --- :( --- :( --- :(

Correct guesses:
Wrong guesses: dipt dipt

continue (q to quit): e
```

Wrong words entered -> losing guesses and the wrong words are now on the "wrong guesses" list.

The players continue playing …

Correct guesses:
Wrong guesses: dipt dipt dipt dipt ghies fmadk sakdfa asagag asadg

continue (q to quit): **a**

@@@ --- @@@ Letters in words @@@ --- @@@
p n i t u d
@@@ --- @@@ --- --- --- --- @@@ --- @@@

You have 1 guesses left
Clues to help you guess
u _ i _
_ u _ _
_ u _ _ i _
i _ _ u _

Enter your guess (4-6 characters): **aada**
:(--- :(--- :(--- :(--- :(--- :(--- :(
Oops ... you lost 1 guess
You have 0 guesses left
:(--- :(--- :(--- :(--- :(--- :(--- :(

The players used up all 10 guesses. The game is over.

The correct and wrong guessed words are displayed.

The total points are displayed.

Correct guesses:
Wrong guesses: dipt dipt dipt dipt ghies fmadk sakdfa asagag asadg aada

:(--- :(--- :(--- :(--- :(
Oops, you used up all your guesses, You got 0 points
Better luck next time
Good Game ... Good Bye

Sample code execution 3 (using WordSet2.txt): **Bold** information entered by a user.

```
|----------------------------------|
|-  LET's Play Word Finder -|
|----------------------------------|

@@@ --- @@@ Letters in words @@@ --- @@@
      i t p d u n
@@@ --- @@@ --- --- --- --- @@@ --- @@@

You have 10 guesses left
Clues to help you guess
u _ i _
_ u _ _
_ u _ _ i _
i _ _ u _

Enter your guess (4-6 characters): pundit
$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$
Yay ... you got 50 points
Total points: 50
$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$

Correct guesses: pundit
Wrong guesses:
obs
continue (q to quit): k

@@@ --- @@@ Letters in words @@@ --- @@@
      n d u i t p
@@@ --- @@@ --- --- --- --- @@@ --- @@@

You have 10 guesses left
Clues to help you guess
u _ i _
_ u _ _
i _ _ u _
_ i _ _

Enter your guess (4-6 characters): unit
$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$
Yay ... you got 10 points
Total points: 60
$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$

Correct guesses: pundit unit
Wrong guesses:

continue (q to quit): k
```

Observe that the clues change from before since the correct word can no longer be used as a clue.

The players continue playing …

```
Correct guesses: pundit unit input punt
Wrong guesses:
continue (q to quit): l
```

@@@ --- @@@ Letters in words @@@ --- @@@
p d t n i u
@@@ --- @@@ --- --- --- --- @@@ --- @@@

You have 10 guesses left
Clues to help you guess
_ i _ _
_ i _ _

Observe that only clues of the unguessed words are displayed. It can be fewer than 4 clues.

Enter your guess (4-6 characters): **dint**
$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$
Yay ... you got 10 points
Total points: 110
$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$

Correct guesses: pundit unit input punt dint
Wrong guesses:
continue (q to quit): **I**

@@@ --- @@@ Letters in words @@@ --- @@@
i n p u d t
@@@ --- @@@ --- --- --- --- @@@ --- @@@

You have 10 guesses left
Clues to help you guess
_ i _ _

Enter your guess (4-6 characters): **pint**
$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$
Yay ... you got 10 points
Total points: 120
$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$

Correct guesses: pundit unit input punt dint pint
Wrong guesses:

The game is over as the players correctly guessed all words in the array (text file).

!!! --- !!! --- !!! --- !!! --- !!!
You got 120 points
Congrats, You completed the list!
Good Game ... Good Bye

Sample code execution 4

a) (using WordSet0.txt): **Bold** information entered by a user.

```
|---------------------------|
|-  LET's Play Word Finder -|
|---------------------------|



 ____    __        __   ____    __  __  ___   ____   ___
( ___)  (  )      /__\ (  _ \  (  \/  )/ __) (  _ \ / __)
 )__)    )(__    /(__)\ )   /   )    ( ( (__   ) _ <\__ \
(____)  (____)  (__)(__)(_)\_) (_/\/\_) \___) (____/ (___/

You have 10 guesses left

Clues to help you guess

_ _ a _
_ _ a _
_ _ _ a _ _ _ e
_ _ a _ _ e _ _

Enter your guess (4-8 characters): scare

$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$
Yay ... you got 20 points
Total points: 20

$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$

Correct guesses: scare

Wrong guesses:
continue (q to quit): h



 ____   ___   ____   ____   ___  __      __  __  __
(  _ \ / __) (  ___) (  _ \ / __)(  )    /__\ (  \/  )
 )   /\__ \  )__)   ) _ <( (__  )(__   /(__)\ )    (
(_)\_)(___/ (____) (____/ \___)(____) (__)(__)(_/\/\_)

You have 10 guesses left
```

b) (using WordSet9.txt): **Bold** information entered by a user.

```
|---------------------------|
|-  LET's Play Word Finder -|
|---------------------------|



 ___    _  _  ____   ____   ___  __  __  ___   ____    __
/ __)  ( )_( )(  _ \ (_  _) / __)(  \/  )/ __) (  _ \  /__\
\__ \   ) _ (  )   /  _)(_ ( (__  )    ( \__ \  )(_) )/(__)\
(___/  (_) (_)(_)\_)(____) \___)(_/\/\_)(___/ (____/(__)(__)

You have 10 guesses left

Clues to help you guess

_ _ a _
_ _ i _
_ _ _ i _ _ _ a _
_ _ a _ _ i _ _ _

Enter your guess (4-9 characters):
```

Sample code execution 5

```
|---------------------------------|
|-  LET's Play Word Finder -|
|---------------------------------|

@@@ --- @@@ Letters in words @@@ --- @@@
      d  a  e  e  r  l  m
@@@ --- @@@ --- --- --- --- @@@ --- @@@

You have 10 guesses left
Clues to help you guess
_ e e _
_ e e _
e _ e _ a _ _
_ e a _ e _
Enter your guess (4-7 characters): mad
Enter your guess (4-7 characters): abcdefghjik
Enter your guess (4-7 characters): redame
:( --- :(  --- :( --- :( --- :( --- :( --- :(
Oops ... you lost 1 guess
You have 9 guesses left
:( --- :(  --- :( --- :( --- :( --- :( --- :(

Correct guesses:
Wrong guesses: redame

continue (q to quit): h
@@@ --- @@@ Letters in words @@@ --- @@@
      e  m  a  r  d  l  e
@@@ --- @@@ --- --- --- --- @@@ --- @@@

You have 9 guesses left
Clues to help you guess
_ e e _
_ e e _
e _ e _ a _ _
_ e a _ e _

Enter your guess (4-7 characters): mader
:( --- :(  --- :( --- :( --- :( --- :( --- :(
Oops ... you lost 1 guess
You have 8 guesses left
:( --- :(  --- :( --- :( --- :( --- :( --- :(

Correct guesses:
Wrong guesses: redame mader
```

continue (q to quit): **l**

@@@ --- @@@ Letters in words @@@ --- @@@
a m r e d l e
@@@ --- @@@ --- --- --- --- @@@ --- @@@

You have 8 guesses left
Clues to help you guess
_ e e _
_ e e _
e _ e _ a _ _
_ e a _ e _

Enter your guess (4-7 characters): **emerald**
$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$
Yay ... you got 50 points
Total points: 50
$$$ --- $$$ --- $$$ --- $$$ --- $$$ --- $$$

Correct guesses: emerald
Wrong guesses: redame mader

continue (q to quit): **q**

|--- G o o d --- T r y ---|
|--- You got 50 points ---|

Unguessed words:
dealer, dermal, leader, marled, medlar, melder, reamed, remade, adeem, alder, ameer, armed, derma, dream, eared, edema, elder, lader, lamed, lamer, laree, madre, medal, merde, merle, ramee, realm, alee, alme, dale, dame, dare, deal, dear, deem, deer, dele, deme, dere, derm, dram, dree, earl, lade, lame, lard, lead, lear, lede, leer, made, male, mare, marl, mead, meal, meed, meld, mere, merl, rale, read, real, ream, rede, reed, reel,

Good Game ... Good Bye