

## Overview

In this project, we will use all the topics we have learn about C to write a program to solve Word Search Puzzles.

## Word Search Puzzle

A word search puzzle is a game that you are asked to find a given list of words from a grid of letters. If you do not know about this puzzle, please visit: [https://en.wikipedia.org/wiki/Word\\_search](https://en.wikipedia.org/wiki/Word_search) to find out more about how this puzzle works.

## Description of wordsearch.c

The main program (`wordsearch.c`) is given to you. Your tasks are to implement 2 functions to complete the program. When the program starts, it will read the puzzle grid from a text file and save it as a 2-D character array. It will also read a list of words from another text file and save the list into a string array. The program will search for words in **5 directions**: horizontal left to right ( $\rightarrow$ ), vertical top to bottom ( $\downarrow$ ), vertical bottom to top ( $\uparrow$ ), diagonal top left to bottom right ( $\searrow$ ), and diagonal bottom left to top right ( $\nearrow$ ).

## Your Tasks

`printPuzzle(char**, int)` – This function takes in the puzzle grid with its dimension (assuming a square grid) and prints out its content. **Implement this function so that it will print out the content in the SAME format as the sample output.**

`searchPuzzle(char**, int, char**, int)` – This function is the core of solving the puzzle. It takes in the puzzle grid with its dimension and tries to find as many words listed in the word list (the 3rd input argument). Not all words will be found from the grid. **Whenever a word is found, your program will print it out as well as converting the word in the puzzle grid to *lower case*.**

**YOU MUST NOT USE ANY ARRAY NOTATION ([]) IN THIS PROGRAM!**

**YOU MUST NOT USE ANY LIBRARY FUNDTIONS TO CONVERT CHARACTERS INTO LOWER CASE!**

Feel free to create any helper functions to simplify your `searchPuzzle` function.

## Testing Your Program

After compiling `wordsearch.c`, run the program by typing `./wordsearch puzzle1.txt`, where `wordsearch` is the executable file, and `puzzle1.txt` is the text file containing the puzzle grid. Feel free to create your own text files for test cases.

## Collaboration

You must credit anyone you worked with in any of the following three different ways:

1. Given help to
2. Gotten help from
3. Collaborated with and worked together

## What to hand in

When you are done with this lab assignment, submit all your work through CatCourses.

***Before*** you submit, make sure you have done the following:

- Your code compiles and runs on a Linux machine (without the need for special libraries).
- Attached `wordsearch.c` and any additional test files.
- Filled in your collaborator's name (if any) in the "Comments..." textbox at the submission page.

Also, remember to demonstrate your code to the TA or instructor before the deadline.