



University of Toronto
APS105H1F — Computer Fundamentals
Lab 5: Arrays and Strings

Introduction

Word Search puzzles are very popular with children and some adults. Being first-year Engineering students, you don't have time to spend on word search puzzles, but after completing this assignment, you can get your computer to solve the puzzles for you while you are busy studying.

In a word search puzzle, you are presented with a table of characters like the following:

```
g z c n c i r c n k s t n g c
j x w v a y r o h q u n b o c
s h f t p m r o t a s i n o b
d i o v a t e q f c r d n l c
x h f a c t r s t s u s d x k
s s u e i e a r p c t d h p g
s e l s t c u o t a v b n c e
l e t p o e d o l q c r b i l
k a w d r z r y g f t e x t i
d w m b a t t e r y d t h e h
r o t s i s e r h l e u j n w
m a u e s l a f e b l r h g h
h m u b o o l i w k s n n a r
l f u w l u f p s f e a m m o
j p x j g e k j m k i p y w x
```

Hidden in this table are the words:

battery bool capacitor char conductor const double electron else false field
float for inductor int magnetic namespace resistor return true void while

The goal when solving word search puzzles is to find all of the given words within the word search puzzle. The words may appear forwards or backwards, horizontally, vertically or along the diagonals of the puzzle. The words battery, false, capacitor and while are more easily seen after their letters are converted to upper case and the puzzle is displayed again as:

```
g z c n C i r c n k s t n g c
j x w v A y r o h q u n b o c
s h f t P m r o t a s i n o b
d i o v A t e q f c r d n l c
x h f a C t r s t s u s d x k
s s u e I e a r p c t d h p g
s e l s T c u o t a v b n c E
l e t p O e d o l q c r b i L
k a w d R z r y g f t e x t I
d w m B A T T E R Y d t h e H
r o t s i s e r h l e u j n W
m a u E S L A F e b l r h g h
h m u b o o l i w k s n n a r
l f u w l u f p s f e a m m o
j p x j g e k j m k i p y w x
```

Your task in this lab assignment is to complete a program that finds words that are hidden in a table of characters. We have prepared a program that provides a few of the first steps towards finishing the task. You need to modify the “starter” program to get it closer to solving word search problems.

Please keep in mind that the marking process will begin approximately forty minutes before the end of the lab.

You should try to complete the lab in advance of your practical session and get help from a TA soon after arriving at SF 1013 so that you may promptly overcome any remaining difficulties.

What to do:

Try to complete the following steps :

1. Using your ECF account:

- (a) Copy the file `/share/copy/aps105/Lab5Starter.c` to your filespace by executing the unix command:

```
cp /share/copy/aps105/Lab5Starter.c ~/Lab5.c
```

- (b) Revise the comments at the top of your `Lab5.c` file so that your name and student number are given.

- (c) Compile the `Lab5.c` program by executing the unix command:

```
g++ Lab5.c -o Lab5
```

- (d) Look at the example word search puzzle and the words that are in the example puzzle by executing the unix commands:

```
more /share/copy/aps105/Lab5Puzzle and
```

```
more /share/copy/aps105/Lab5Words
```

- (e) Run the `Lab5` program to confirm that it works by executing the unix command:

```
./Lab5
```

Enter the filename `APS105` to use the example word search puzzle. Try searching for the word `battery`. The program should report:

```
The search word is: battery
```

```
.... Searching ....
```

```
The word was found in the puzzle!
```

- (f) Once you have finished the above steps, you are ready to update the program to make it more complete. If you are developing your program on your own computer, you will want to copy the `Lab5Starter.c` file, the example word search puzzle file (`/share/copy/aps105/Lab5Puzzle`) and the words file (`/share/copy/aps105/Lab5Words`) to your computer. You will need to appropriately modify the `defaultFilename` variable in the `readPuzzle()` function to reflect the fact that the example puzzle file is stored on your computer. Remember to reset the variable when you copy your program back to ECF for your lab period.

2. Complete the `printPuzzle()` function by following these steps:

- (a) Revise the program in your `Lab5.c` file so that the `printPuzzle()` function outputs a readable display of the word search puzzle. The display format should be similar to the display format used earlier in this handout.

- (b) Compile and run your program. Enter the filename `APS105` to use the example word search puzzle. Try searching for the word `battery`. Your program should display the whole word search puzzle. You should notice the word `BATTERY` displayed in the puzzle.
3. Add a function to the `Lab5.c` file that finds search words when they appear backwards in a row of the puzzle by following these steps:
 - (a) Read the body of the function `checkForwardHorizontal()` **very** carefully. Do not proceed until you understand how it works. Try tracing the instructions in the function.
 - (b) Create a copy of the whole function `checkForwardHorizontal()` in your `Lab5.c` file.
 - (c) Rename your new copy of the `checkForwardHorizontal()` function and revise it so that it finds search words that appear backwards in the search word puzzle. You will also need to declare your new function near the top of the `Lab5.c` file. And, of course, you will need to add appropriate instruction(s) that call your new function from the `main()` function.
 - (d) Test your program. It should find that the word `false` appears backwards on line 12 of the example puzzle.
4. Add a function to the `Lab5.c` file that finds search words when they appear vertically down a column of the puzzle by following these steps:
 - (a) Create another copy of the whole function `checkForwardHorizontal()` in your `Lab5.c` file.
 - (b) Rename your new copy of the `checkForwardHorizontal()` function and revise it so that it finds search words that appear vertically down a column in the search word puzzle. You will also need to declare your new function near the top of the `Lab5.c` file. And, of course, you will need to add appropriate instruction(s) that call your new function from the `main()` function.
 - (c) Test your program. It should find that the word `capacitor` appears downwards from line 1 of the example puzzle.
5. Add a function to the `Lab5.c` file that finds search words when they appear vertically up a column of the puzzle by following these steps:
 - (a) Create another copy of the whole function `checkForwardHorizontal()` in your `Lab5.c` file.
 - (b) Rename your new copy of the `checkForwardHorizontal()` function and revise it so that it finds search words that appear vertically up a column in the search word puzzle. You will also need to declare your new function near the top of the `Lab5.c` file. And, of course, you will need to add appropriate instruction(s) that call your new function from the `main()` function.
 - (c) Test your program. It should find that the word `while` appears upwards from line 11 of the example puzzle.

After completing the above steps, your lab solution will be considered complete.

If you were not challenged by the steps, you may earn a **bonus** of at most 2 marks out of 10 by revising your program so that it finds words that appear forwards or backwards along the diagonals of the search word puzzle. Your program should be able to find the search words `double`, `true`, `inductor` and `field` along diagonals in the example word search puzzle.

Finally, if you have time and are interested, visit the website puzzlemaker.school.discovery.com/WordSearchSetupForm.html

and have fun making up new word search puzzles. When you store a puzzle in a file to be read into your Lab5 program, you will need to remove all of the blanks at the start of the rows in the puzzle.

Requirements

Your program must be in a file named `Lab5.c`. There should be additional meaningful comments in your source code.

You must be able to compile your program on the ECF system and demonstrate it for the TA who marks your lab.

Submission

Before your lab is over you **must** submit your file using the unix `submitaps105f` command. Use the command `submitaps105f 5 Lab5.c` to submit your file. The manual page for this command may be read by executing the unix `man submit` command.

The command `submitaps105f -l 5` may be used to confirm that you have successfully submitted your file.