

COSC 2436 lab1: Word Search with Recursion

Create a C++ program to check to see if a string exists in a given matrix of characters using recursion. Your program should read the dimensions of the character matrix from an input file. It should then read the matrix and load it into a dynamically allocated 2D-array. Finally, it should read the string to be found. Output the word true if the string can be constructed from characters in the matrix, and the word false otherwise.

The string can be constructed from a sequence of character of adjacent index, where adjacent an index is connected horizontally or vertically, but not wrapping around edges.

1. Input files

- The first line of the input will contain two integers 'm' and 'n', separated by a single space. 'm' represents the number of rows and 'n' represents the number of columns.
- The following 'm' lines will each contain 'n' number of characters, each character will be separated by a single space.
- The last line in the input will contain the string to be searched in the matrix.
- Each character in the matrix can only be used once, meaning that once a character has been visited, it cannot be visited again.
- All character and string in the input will be in lowercase
- There will be no extra lines and whitespaces in the input.

2. Output files

- Output the word true if the string exists in the matrix; otherwise output the word false.

3. Examples

input1.txt

```
5 5
m h j f n
r j e u l
y p l w b
a p f t j
r i s v z
apple
```

output1.txt

true

input2.txt

```
4 4
a y u n
o b j k
g i c w
p h d q
abc
```

output2.txt

false

COSC 2436 lab1: Word Search with Recursion

input3.txt

output3.txt

```
7 10
c v c o e o i l j q
w q f a b e g k c q
v d a v i c l t a u
e c w m o c t s p i
m l s p u a c z n j
i t a b t k t y i n
c l v r e h t p p y
computer
```

true

4. Reminder

- Turn in your lab assignment to our Linux server, follow the link [here](#) for more instructions.
- Make sure to only have **one (1)** .cpp file with the main() function in your working directory, otherwise your program will fail the grading script.
 - Create a folder under your root directory, name the folder *lab1* (case sensitive), copy all your .cpp and .h files to the folder (ArgumentManager.h is also needed)
 - Only include the necessary files (.cpp and .h files) in your working directory in your final submission
 - To test your program, copy the input files into the server and run your program. After verifying that they pass, delete the .txt files.

Please reach out to myself or the TAs for any clarifications or typos.