CO222-2022- Crossword Puzzle Solver

Marks:
20% of the final grade (zero marks for submissions that have compilation errors)

Deadline:

Phase I [10%]: strictly on or before Jan 10th @ 11.55 pm. Late submissions will be accepted with a 20% penalty per day they are late.

Phase II [10%]: strictly on or before Jan 17th @ 11.55 pm. Late submissions will be accepted with a 20% penalty per day they are late.

Plagiarism policy:
Copied submissions (including those from the Internet) will receive zero marks. Your program must be entirely your own work. Do not copy from others, and do not allow anyone else to see your code.

Task:
For this project, you will write a C program to solve a crossword puzzle. Your program should read, from standard input, a puzzle grid and a set of words and should print the solved puzzle to standard output.

Specification:
A partially filled puzzle grid will be given in the input with stars, hashes and letters. Stars indicate blocked cells (cannot be filled). Hashes indicate vacant cells needed to be filled with letters. The inputs will also include the words to be used, one per line, in random order. If your program finds out that filling the crossword puzzle is impossible with the given words, you should print IMPOSSIBLE.

Input:
The grid is followed by an empty line and a list of words, one per line. To finish giving words, an empty line is provided.

Output:
The completed puzzle, the message "IMPOSSIBLE" or "INVALID INPUT".

For example,

```
./puzzle
*#**
####
*#**
****

FLY
GLUE

*F**
GLUE
*Y**
****
```

Remember that your program is going to be **automatically** marked for functional correctness. Therefore, be careful to produce EXACTLY the desired output. Most testing will be on legal input; however, your program may be tested on invalid input. When you determine its invalid input, you print "INVALID INPUT".

Test cases:

Here are some test cases. This is NOT the complete set of test cases used to evaluate your submission. Therefore, you are advised to formulate your test cases (in addition to what is given here) and test your program.

Testcase 01:

Input :

```
*#**
####
*#**
****

Cricket
```

Expected Output:

```
IMPOSSIBLE
```

Testcase 02:

Input :

```
****
####
****
*###

FIRE
CAT
```

Expected Output:

```
****
FIRE
****
*CAT
```

Testcase 03:

Input :

```
*#**
####
*#**
****

FLY
GLUE
```

Expected Output:

```
*F**
GLUE
*Y**
****
```

Submission:

You need to solve the problem using two different methods, static memory, and dynamic memory.

Phase I: Using static allocation of arrays. Name this file **puzzle-static.c**

Phase II: Using dynamic memory allocation. Name this file **puzzle-dynamic.c**. Prepare a report to compare the execution time and memory space usage differences between the two programs. Name this file **puzzle-compare-report.pdf**

Note that, marks will also be awarded for legible and readable code and proper comments.

Submit the files (one file for Phase I and two files for Phase II) in the corresponding links in FEeLS.