# COP 3503C  Programming Assignment # 2

## BackTracking

## Read all the pages before starting to write your code

**What should you submit?**

Write all the code in a single Main.java file and **Lastname_analysis.txt**. The analysis file will briefly discuss the time complexity of your code. **Submit your** Main.java **to Mimir platform.**

Please include the following commented lines in the beginning of your code to declare your authorship of the code:

/* COP 3503C Assignment 2

This program is written by: Your Full Name */

**Compliance with Rules:** UCF Golden rules apply towards this assignment and submission. Assignment rules mentioned in syllabus, are also applied in this submission. The TA and Instructor can call any students for explaining any part of the code in order to better assess your authorship and for further clarification if needed.

Caution!!!

Sharing this assignment description (fully or partly) as well as your code (fully or partly) to anyone/anywhere is a violation of the policy. I may report to office of student conduct and an investigation can easily trace the student who shared/posted it. Also, getting a part of code from anywhere will be considered as cheating.

## Deadline:

See the deadline in Mimir. The assignment will accept late submission up to 24 hours after the due date time with 10% penalty. After that the assignment submission will be locked. An assignment submitted by email will not be graded and such emails will not be replied according to the course policy.

**What to do if you need clarification on the problem?**

I will create a discussion thread in webcourses and I highly encourage you to ask your question in the discussion board. Maybe many students might have same question like you. Also, other students can reply and you might get your answer faster. Also, you can write an email to the TAs and put the course teacher in the cc for clarification on the requirements.

**How to get help if you are stuck?**

According to the course policy, all the helps should be taken during office hours. Occasionally, we might reply in email.

# Problem: Our Word Finder

Given a MxN matrix of characters and a string word, your task will be to find the word in the given matrix. The rule is:

1. The word can be formed in any direction such as:
   a. Left to right
   b. Right to left
   c. Left to right diagonal
   d. Right to left diagonal
   e. Any combination above
   f. Same letter cannot be re-used to form the word
   g. Must follow any one of the above directions and their combination while forming the word.

Example:

Consider the following 3X4 matrix.

| C | B | A | B |
|---|---|---|---|
| B | N | O | D |
| M | D | E | E |

*Let us try to find the following words:*

*"ABC" is available in the matrix:*

| C | B | A | B |
|---|---|---|---|
| B | N | O | D |
| M | D | E | E |

*BAOED is available in the matrix:*
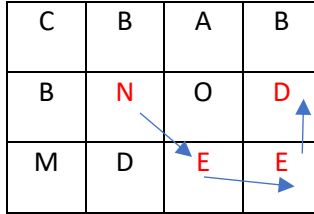
| C | B | A | B |
|---|---|---|---|
| B | N | O | D |
| M | D | E | E |

*CNE is available in the matrix:*

| C | B | A | B |
|---|---|---|---|
| B | N | O | D |
| M | D | E | E |

*NEC is not available in the matrix*

*Need is available in the matrix:*

| C | B | A | B |
|---|---|---|---|
| B | N | O | D |
| M | D | E | E |

*NEEDON is not available in the matrix.*

## Input File Format (Your code must read from a file called in.txt as we will also pass in.txt file to your code while grading)

The first line of the file will have a single positive integer k, representing the number of test cases in the file. Then there will be k groups of inputs representing k different test cases.

The first line of a test case consists of 3 integers, M, N, and S. Here M = number of rows in the matrix, N = numer of columns in the matrix, and S = number of words you need to find from this matrix.

Then M lines represent the rows input characters, were each line has N characters saperated by space.

After M lines, there will be S lines of words. You need to find these words in the matrix.

Similarly, K test cases will be provided in the above format.

Here's an example in.txt file:

```
2
3 4 6
C B A B
B N O D
M D E E
ABC
BAOED
CNE
NEC
NEED
NEEDON
4 4 4
c o m p
a m a p
a p s s
x y z q
mop
comp
omo
maa
```

## Output Format (Your must have to use standard console output to display your result)

Please get an idea of the output format from the following output for the above input file.

```
Test#1
Looking for ABC
[C, B, A,  ]
[ ,  ,  ,  ]
[ ,  ,  ,  ]

Looking for BAOED
[ , B, A,  ]
[ ,  , O,  ]
[ , D, E,  ]

Looking for CNE
[C,  ,  ,  ]
[ , N,  ,  ]
[ ,  , E,  ]

Looking for NEC
NEC not found!
Looking for NEED
[ ,  ,  ,  ]
[ , N,  , D]
[ ,  , E, E]

Looking for NEEDON
NEEDON not found!

Test#2
Looking for mop
mop not found!
Looking for comp
[c, o, m, p]
[ ,  ,  ,  ]
[ ,  ,  ,  ]
[ ,  ,  ,  ]

Looking for omo
omo not found!
Looking for maa
[ ,  ,  ,  ]
[a, m,  ,  ]
[a,  ,  ,  ]
[ ,  ,  ,  ]
```

**Specific Restrictions:**

**Your code must incorporate the following restrictions to receive full credit:**

1. You must have to use backtracking strategy to solve this problem
2.

**Submission:**

**Submit the following files in mimir. Make sure your code work in mimir platform.**

1. Main.c file that process your file

2. **Lastname_analysis.txt file:** This file will very briefly explain the time complexity of your code

# Rubric (subject to change):

According to the Syllabus, the code will be compiled and tested in Mimir Platform for grading. If your code does not compile in Mimir, we conclude that your code is not compiling and it will be graded accordingly. We will apply a set of test cases to check whether your code can produce the expected output or not. Failing each test case will reduce some grade based on the rubric given bellow. If you hardcode the output, you will get -200% for the assignment.

1. If a code does not compile, the code may get 0. However, some partial credit maybe awarded. A code having compiler error cannot get more than 35% even most of the codes are correct
2. Coding and applying backtracking strategies: 35%
3. Passing test cases (exact output format): 55%
4. Reading data from file properly: 5%
5. Analysis file with correct content: 5%

Penalty:

6. Not using backtracking: -60%
7. Not putting header comment: -10%
8. Poor indentation in code and spacing in code: -10%
9. Not putting proper comments on important block of code: -5% (don't start commenting every line. Only important block of code)
10. Not reading from in.txt file: -10%

## Good Luck!