



Practice Arena

Practice problems aimed to improve your coding skills.

- 📁 PRACTICE-02_SCAN-PRINT
- 📁 PRACTICE-03_TYPES
- 📁 LAB-PRAC-02_SCAN-PRINT
- 📁 LAB-PRAC-01
- 📁 PRACTICE-04_COND
- 📁 BONUS-PRAC-02
- 📁 LAB-PRAC-03_TYPES
- 📁 PRACTICE-05_COND-LOOPS
- 📁 LAB-PRAC-04_COND
- 📁 LAB-PRAC-05_CONDDLOOPS
- 📁 PRACTICE-07_LOOPS-ARR
- 📁 LAB-PRAC-06_LOOPS
- 📁 LAB-PRAC-07_LOOPS-ARR
- 📁 LABEXAM-PRAC-01_MIDSEM
- 📁 PRACTICE-09_PTR-MAT
- 📁 LAB-PRAC-08_ARR-STR
- 📁 PRACTICE-10_MAT-FUN
- 📁 LAB-PRAC-09_PTR-MAT
 - ❓ Mr C writes a Story
 - ❓ Matrix Arithmetic
 - ❓ Spin the Matrix
 - ❓ Crony Capitalization
 - ❓ Matrix Mirroring
 - ❓ Sodoku
 - ❓ The Last Line
 - ❓ Singular Value Decomposition
 - ❓ Matrix Flip
 - ❓ Now we are in Rome
 - ❓ Search for the Submatrix
 - ❓ Convoluted Convolutions
- 📁 LAB-PRAC-10_MAT-FUN
- 📁 PRACTICE-11_FUN-PTR
- 📁 LAB-PRAC-11_FUN-PTR
- 📁 LAB-PRAC-12_FUN-STRUC
- 📁 LABEXAM-PRAC-02_ENDSEM
- 📁 LAB-PRAC-13_STRUC-NUM
- 📁 LAB-PRAC-14_SORT-MISC

Search for the Submatrix

LAB-PRAC-09_PTR-MAT

Search for the Submatrix [20 marks]

Problem Statement

Given a string, any contiguous set of characters occurring in that string is considered a substring of that string. Similarly we can extend that notion to submatrices as well. In the first line of the input, you will be given two strictly positive integers n and m , separated by a space. In the next n lines, you will be given the entries of an $n \times m$ integer matrix A with one row of A in each line with two entries separated by a single space.

Then in the next line you will be given two strictly positive integers k and l , separated by a space. In the next k lines, you will be given the entries of a $k \times l$ integer matrix B with one row of B in each line with two entries separated by a single space. We promise that k will be less than or equal to n and l will be less than or equal to m .

In your output you have to tell us if the matrix B occurs as a submatrix in A or not. If B never occurs as a submatrix inside A , simply print the words "SUBMATRIX NOT PRESENT" (without quotes) in the output. However, if B is present one or more times in A as a submatrix, then you have to print the indices (as in 2D array indices) of the top left hand corners of all these occurrences in the output, one occurrence in each line in a format given below.

Make sure you first output all occurrences (if any) where the top left hand corner is in the first row of A , followed by all occurrences (if any) where the top left hand corner is in the second row of A , followed by all occurrences (if any) where the top left hand corner is in the third row of A and so on and so forth. If there are multiple occurrences with top left hand corner on the same row of A , output these occurrences in increasing order of the column number (index) of the top left hand corner.

Caution

1. We will not penalize you if there are extra newlines at the end of your output.
2. However, there should not be any stray, trailing spaces in your output.
3. Note that if your occurrences are given in the wrong order, you may end up getting zero by the autograder even if you output all occurrences correctly. This is because in order to get partial credit (see grading scheme below), you must output the correct occurrence in the correct order.
4. Submatrix occurrences may overlap with each other, just as substrings may overlap with each other.

Code to manipulate matrices

```
int m, n;
scanf("%d %d", &m, &n);
int num[m][n], i, j;
for(i = 0; i < m; i++)
    for(j = 0; j < n; j++)
        scanf("%d", &num[i][j]);
        printf("%d", num[i][j]);
```

EXAMPLE:

INPUT

2 4

1 5 1 5

2 6 1 5

1 2

1 5

OUTPUT:

(0, 0)

(0, 2)

(1, 2)

Explanation: Note that there were three occurrences of B in A but the two occurrences in the first row got reported first and then the occurrence in the second row got reported. Also, within the two occurrences in the first row, the one with smaller column number got reported first.

Grading Scheme:Total marks: **[20 Points]**

There will be partial grading in this question. There may be several lines in your output. Printing each line correctly, in the correct order, carries equal weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

 **Start Solving!** (/editor/practice/6192)