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_CONDLOOPS
- 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

Singular Value Decomposition

LAB-PRAC-09_PTR-MAT

Singular Value Decomposition [20 marks]

Problem Statement
Every m x n real matrix A can be decomposed as A = U * S * V' (where ' indicates the transpose operation) where U is an m x m unitary matrix (i.e. U is its own inverse), S is an m x n diagonal matrix (all off-diagonal entries of S, i.e. entries of the form S_{ij} where $i \neq j$ are zero. Only entries of the form S_{ii} can be non-zero) and V is an n x n unitary matrix (i.e. V is its own inverse).

Both m and n are guaranteed to be less than or equal to 100. In the first line of the input, you will be given m and n as two strictly positive integers, separated by a single space. In the next line, you will be given the diagonal entries of S in a single line (two entries separated by a space). In then next m lines, you will be given the m rows of the matrix U and in the next n lines you will be given the n rows of the matrix V. U, S, V will have only integer entries.

You have to output the matrix A = U * S * V' as your output. Output one row of A in each line with a single space between two entries of that row. Make sure there are no trailing spaces at the end of each line, as well as no trailing new lines in your output.

Caution

1. Be careful about extra/missing lines and extra/missing spaces in your output.
2. V appears transposed in the expression for A
3. Be careful to count how many diagonal entries, a rectangular matrix has.

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
3 2
2 2
1 0 0
0 1 0
0 0 1
1 0
0 1

OUTPUT:
2 0

0 2
0 0

Explanation: In this case, the matrix S will look like
2 0
0 2
0 0
Note that only diagonal entries are non-zero.

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

There will be partial grading in this question. There will be m 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/6189)