

Sabancı University

Faculty of Engineering and Natural Sciences
CS204 Advanced Programming
Fall 2016-2017

Homework1 - Matrix Search

Due: 04/10/2016, 23:00

PLEASE NOTE:

Your program should be a robust one such that you have to consider all relevant programmer mistakes and extreme cases; you are expected to take actions accordingly!

**You HAVE TO write down the code on your own.
You CANNOT HELP any friend while coding.
Plagiarism will not be tolerated!**

Introduction: The aim of this homework is to recall CS201 material and practice on matrices (two dimensional arrays/vectors). You are asked to find the number of distinct letters in valid sub-matrices of an input matrix and output the number and the letters themselves. The details about the homework will be explained in the following sections of this document.

Input: The program prompts for the input file name. Then, it reads the file name from the standard input. A simple input file can be as follows:

a	b	C
A	d	c

Sample input file

Notice that the input file only contains the letters a-zA-Z. You have to check; 1) each line has the same number of characters, 2) the file contains only lower- or upper-case letters and nothing else. Your program should be able to detect cases where the input file is invalid. You have to check and report any irregularities in the input file. If the file is valid you need to use an appropriate container to hold this 2-dimensional matrix in memory. Please see sample runs to see some irregularities that may occur in the input files.

Please read the definitions and instructions very carefully. THEY ARE IMPORTANT if you want to submit a complete and correct homework. In addition, after each homework assignment, a forum will be added in the SUCourse / Discussions part. All students are expected to ask their homework related questions in that forum and get further explanations.

Finding the number of distinct letters in **sub-matrices**:

- 1- Let the original matrix **M** be an **R**×**C** letter matrix where **R** is the number of rows and **C** is the number of columns in **M**. The rows of **M** are numbered in a top-to-bottom fashion; that is the top row is the first row and the bottom row is the **R**'th row. Similarly, the columns of **M** are numbered in a left-to-right fashion, i.e., the left column is numbered as the first one.
- 2- A sub-matrix is defined by a triplet **((x,y), r, c)** where **x** and **y** are the row and column ids of the top-left entry of the sub-matrix, and **r** and **c** are its number of rows and columns, respectively. Notice that a triplet is a valid sub-matrix if **x + r - 1 ≤ R** and **y + c - 1 ≤ C** (see Figure 1). **The original matrix itself can also be a sub-matrix.** For instance, the matrix in the sample input file (page 1) is a sub-matrix defined by the triplet **((1,1), 2, 3)**.

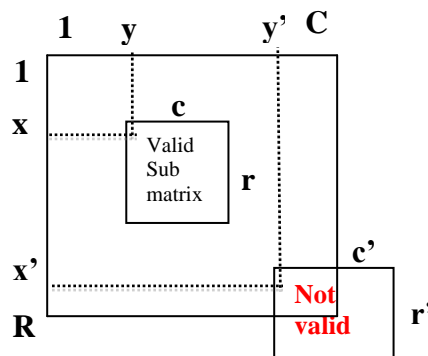


Figure 1

- 3- You are expected to search original matrix and find the number of *distinct* letters in all its sub-matrices. For instance, the letters 'a' and 'b' are distinct. However, in this homework the letters 'a' and 'A' are considered the same. Hence, the sample input matrix above (which is also a valid sub-matrix with triplet **((1,1), 2, 3)**) contains 4 distinct letters (a,b,c,d).
- 4- In the output, the information for each sub-matrix must be reported in a different line. At each line you need to report the following: **1)** the triplet, **2)** the number of distinct letters and **3)** the distinct (**all lower-case**) letters themselves. All these need to be **separated with a colon ':'**. That is you need to output **((x,y),r,c):#letters:letters** for each sub-matrix. Suppose we have the previous example in the first page as the original matrix **M**:

Sample outputs for a couple of sub-matrices are below:

Sub-matrix	Output format
a	((1,1),1,1):1:a Left-top and right-bottom elements are same.
A	((2,1),1,1):1:a Left-top and right-bottom elements are same.
a A	((1,1),2,1):1:a (1,1) is the left-top cordinate 2 rows -1 column
$b \ C$ $d \ c$	((1,2),2,2):3:bcd (1,2) is the coordinate of left-top element, i.e., the coordinate of 'b' in original matrix. # of distinct letters are 3 (since C and c are counted once)
$d \ c$	((2,2),1,2):2:cd (2,2) is the coordinate of d in original matrix
$a \ b \ C$	((1,1),1,3):3:abc

- 5- **The output MUST BE SORTED in the order of x, y, r, c.** For instance, all the sub-matrices starting with an entry in the first row of M must be reported before the others. Among these sub-matrices, the ones with a lower y should be reported first. And among these, the ones with less number of rows must be reported before the others.

Sample executions and corresponding output are given below:

File: input1.txt

A bC 2dc

Output:

Enter the name of the file

input1.txt

Error: Matrix contains invalid letter: 2

Program Exiting...

Press any key to continue . . .

File: input2.txt

a b
B dc

Output:

Enter the name of the file

Input2.txt

Error: number of element in each rows are not same

Program Exiting...

Press any key to continue . . .

File: input3.txt

a b
AB

Output:

Enter the name of the file

Input3.txt

Here are the distinct letters for each sub-matrix

```
((1,1),1,1):1:a
((1,1),1,2):2:ab
((1,1),2,1):1:a
((1,1),2,2):2:ab
((1,2),1,1):1:b
((1,2),2,1):1:b
((2,1),1,1):1:a
((2,1),1,2):2:ab
((2,2),1,1):1:b
```

Press any key to continue . . .

File: input4.txt

abC
Adc

Output:

Enter the name of the file

Input4.txt

Here are the distinct letters for each sub-matrix

```
((1,1),1,1):1:a
((1,1),1,2):2:ab
```

```

((1,1),1,3):3:abc
((1,1),2,1):1:a
((1,1),2,2):3:abd
((1,1),2,3):4:abcd
((1,2),1,1):1:b
((1,2),1,2):2:bc
((1,2),2,1):2:bd
((1,2),2,2):3:bcd
((1,3),1,1):1:c
((1,3),2,1):1:c
((2,1),1,1):1:a
((2,1),1,2):2:ad
((2,1),1,3):3:acd
((2,2),1,1):1:d
((2,2),1,2):2:cd
((2,3),1,1):1:c
Press any key to continue . . .

```

File: input5.txt

a bC D Adc e

Output:

Enter the name of the file

Input5.txt

Here are the distinct letters for each sub-matrix

```

((1,1),1,1):1:a
((1,1),1,2):2:ab
((1,1),1,3):3:abc
((1,1),1,4):4:abcd
((1,1),2,1):1:a
((1,1),2,2):3:abd
((1,1),2,3):4:abcd
((1,1),2,4):5:abcde
((1,2),1,1):1:b
((1,2),1,2):2:bc
((1,2),1,3):3:bcd
((1,2),2,1):2:bd
((1,2),2,2):3:bcd
((1,2),2,3):4:bcde
((1,3),1,1):1:c
((1,3),1,2):2:cd
((1,3),2,1):1:c
((1,3),2,2):3:cde
((1,4),1,1):1:d
((1,4),2,1):2:de
((2,1),1,1):1:a

```

```
((2,1),1,2):2:ad
((2,1),1,3):3:acd
((2,1),1,4):4:acde
((2,2),1,1):1:d
((2,2),1,2):2:cd
((2,2),1,3):3:cde
((2,3),1,1):1:c
((2,3),1,2):2:ce
((2,4),1,1):1:e
Press any key to continue . . .
```

Some Important Rules:

In order to get a full credit, your programs must be efficient and well presented, presence of any redundant computation or bad indentation, or missing, irrelevant comments are going to decrease your grades. You also have to use understandable identifier names, informative introduction and prompts. Modularity is also important; you have to use functions wherever needed and appropriate.

When we grade your homeworks we pay attention to these issues. Moreover, in order to observe the real performance of your codes, we may run your programs in *Release* mode and **we may test your programs with very large test cases.**

What and where to submit (PLEASE READ, IMPORTANT): You should prepare (or at least test) your program using MS Visual Studio 2012 C++. We will use the standard C++ compiler and libraries of the abovementioned platform while testing your homework. It'd be a good idea to write your name and last name in the program (as a comment line of course).

Submissions guidelines are below. Some parts of the grading process are automatic. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.

Name your cpp file that contains your program as follows:

“SUCourseUserName_YourLastname_YourName_HWnumber.cpp”

Your SUCourse user name is actually your SUNet username that is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and last name is Özbugsizkodyazaroglu, then the file name must be:

Cago_Ozbugsizkodyazaroglu_Caglayan_hw2.cpp

Do not add any other character or phrase to the file name. Make sure that this file is the latest version of your homework program. Compress this cpp file using WINZIP or WINRAR programs. Please use "zip" compression. "rar" or another compression mechanism is NOT allowed. Our homework processing system works only with zip files. Therefore, make sure that the resulting compressed file has a zip extension. Check that your compressed file opens up correctly and it contains your cpp file.

You will receive no credits if your compressed zip file does not expand or it does not contain the correct file. The naming convention of the zip file is the same as the cpp file (except the extension of the file of course). The name of the zip file should be as follows:

SUCourseUserName_YourLastname_YourName_HWnumber.zip

For example zubzipler_Zipleroglu_Zubeyir_hw1.zip is a valid name, but

hw1_hoz_HasanOz.zip, HasanOzHoz.zip

are **NOT** valid names.

Submit via SUCourse ONLY! You will receive no credits if you submit by other means (e-mail, paper, etc.).

Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

Good Luck!

CS204 Team (Leyli Javid Khayati)