Sabancı University

Faculty of Engineering and Natural Sciences

CS204 Advanced Programming
Fall 2019-2020
Homework 1 – Matrix Operations
Due: 24/09/2019 - 11:55pm
(NO LATE SUBMISSION)

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!

1. Introduction

The aim of this homework is to recall CS201 material and practice on matrices (two dimensional arrays/vectors) and file input/output. You are asked to implement a program which finds counts of words(strings) with given N characters in a 2D matrix. The details of the homework will be explained in the following sections of the document.

2. Input

The program prompts for the input file name. Then, it reads the file name from the standard input. The first line of the input file contains **a single integer** stating the **number of rows in the matrix (M)**, which is equal to the **number of columns** because matrices will be **square matrix** (M x M matrix with M rows and M columns).

The following lines will be the rows of matrix. All elements in the cells of matrix **must be** uppercase letters (A-Z) – your program must check this. You also need to check if there are indeed M rows and M columns. There is an empty line after matrix rows. At the following line of empty line, there is a single integer (**N**) for the length of strings you will be searching in the matrix. A sample input file can be as follows:

```
4
ABCD
BCDA
CDAB
DABC

3
```

Program Flow

What your program needs to do is find and list strings with given length **N** and the number of their appearances in the **(M x M)** matrix.

- The strings can start from any cell in the matrix.
- Once you are in a cell, you can go any of the four directions: left, right, up and down.
- You must **NOT** visit the same cell while generating a string.
- You will stop once you have **N** letters in your string (i.e., once you visit **N** cells).

Boundaries of variables are as follows: **0 < N < 20** and **0 < M < 20**. With that in mind, the flow of the program will be as following:

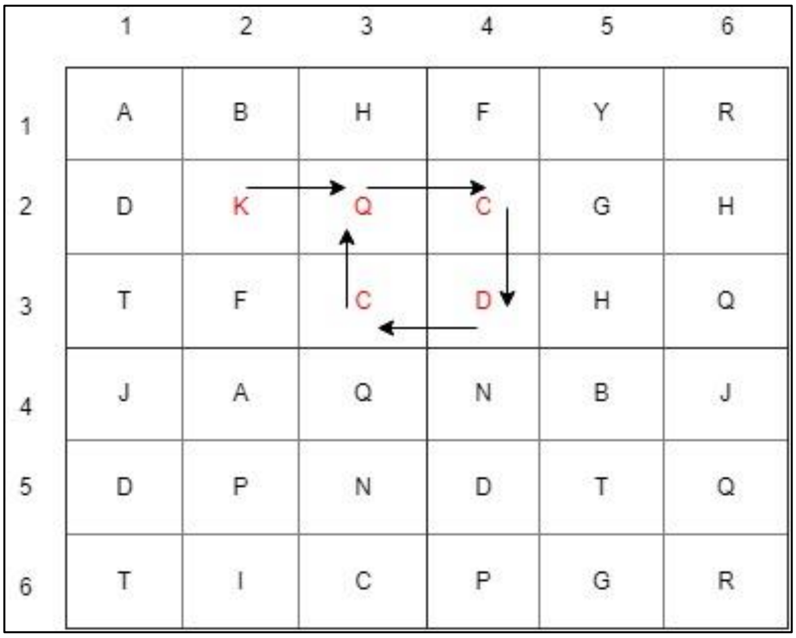
- Prompt for the input file name and perform an input check. If an invalid file name is entered, ask again until a correct file name is entered.
    - Read the value of **M** from the file. Store the matrix in the given input file. If the size of matrix does not satisfy the given row (equal to column) value **M**, print an error message and end the execution.
- Read **N** from input file.
- Find all the strings of length **N** in the matrix. Do not forget: while generating strings, the directions to move are left, right, up, down.
- Store the strings you have found and the number of times they exist in the matrix.
- Print out the strings and the number of their appearances. For the output format check sample runs.

Sample output for the sample input given above is as following:

```
The given matrix includes:

[+] ABC exists 8 many times.

[+] BCD exists 8 many times.

[+] BAB exists 4 many times.

[+] BCB exists 4 many times.

[+] CDA exists 10 many times.

[+] CBC exists 4 many times.
[+] CDC exists 4 many times.
[+] CBA exists 8 many times.
[+] DAB exists 8 many times.
[+] DCD exists 6 many times.
[+] DAD exists 6 many times.
[+] DCB exists 8 many times.
[+] ADA exists 4 many times.
[+] ABA exists 4 many times.
[+] ADC exists 10 many times.
[+] BAD exists 8 many times.
```

Note:

Reuse of same cell is not allowed while generating strings. The following image is an example to the rule where the size of the strings **N** is set to 6.



	1	2	3	4	5	6
1	A	B	H	F	Y	R
2	D	K	Q	C	G	H
3	T	F	C	D	H	Q
4	J	A	Q	N	B	J
5	D	P	N	D	T	Q
6	T	I	C	P	G	R

It is not allowed to generate **“KQCDCQ” as in the figure** since Q at (2,3) is visited twice. However the string **“KQCDCQ” is in the matrix:** the last Q can come from (4,3). So your program must report only this.

3. Sample Runs

Below you can find sample outputs for the program:

input.txt

```
4
ABC
BCDDF
CDAB
DABC

3
```

Case 1:

```
Please enter the input file name: input1.txt
Couldn’t open file. Please enter again: input.txt
Input file is not as expected.
Press any key to continue . . .
```

input2.txt

```
4
ABCD
BCDA
CDAB

3
```

Case 2:

```
Please enter the input file name: input.txt
Input file is not as expected.
```

```
Press any key to continue . . .
```

Input3.txt

```
3
AB*
BCD
CDA


3
```

Case 3:

```
Please enter the input file name: input3.txt
Input file is not as expected.
Press any key to continue . . .
```

Input4.txt

```
3
555
BCD
CDA


3
```

Case 4:

```
Please enter the input file name: input4.txt
Input file is not as expected.
Press any key to continue . . .
```

Input5.txt

```
4
ABCD
BCDA
CDAB
DABC

3
```

Case 5:

```
Please enter the input file name: input5.txt
Given matrix includes:
[+] ABC exists 8 many times.
[+] BCD exists 8 many times.
[+] BAB exists 4 many times.
[+] BCB exists 4 many times.
[+] CDA exists 10 many times.
[+] CBC exists 4 many times.
[+] CDC exists 4 many times.
[+] CBA exists 8 many times.
[+] DAB exists 8 many times.
[+] DCD exists 6 many times.
[+] DAD exists 6 many times.
[+] DCB exists 8 many times.
[+] ADA exists 4 many times.
[+] ABA exists 4 many times.
[+] ADC exists 10 many times.
[+] BAD exists 8 many times.
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 valent, name is Valentina, and last name is Tereşkova, then the file name must be:

Valent_Tereskova_Valentina_hw1.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 (M. Yusa Erguven, Kamer Kaya)