Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 3 REPORT

NURULLAH GÜNDOĞDU
161044108

ÖZGÜ GÖKSU

1 INTRODUCTION

1.1 Problem Definition

1.1.1 Part1

Bu bölümde 1'ler ve 0'lardan oluşmuş bir matrisi stack yardımıyla matris içinde kaç sayı grubu bulmamız gerekmektedir. Matris boyutu bilinmemektedir. Kullanıcı istediği boyutta bir matris girebilir. In this section, we need to find the number of numbers in the matrix with the help of a matrix stack consisting of 1s and 0's. The matrix size is unknown. The user can enter a matrix of the desired size.

1.1.2 Part2

In this section, we are asked to print the result of the equation in the input file entered by the user. There are equations and parameters in the file. Parameters can be entered optionally. It is desirable that we use the values of the entered parameters in equations.

1.2 System Requirements

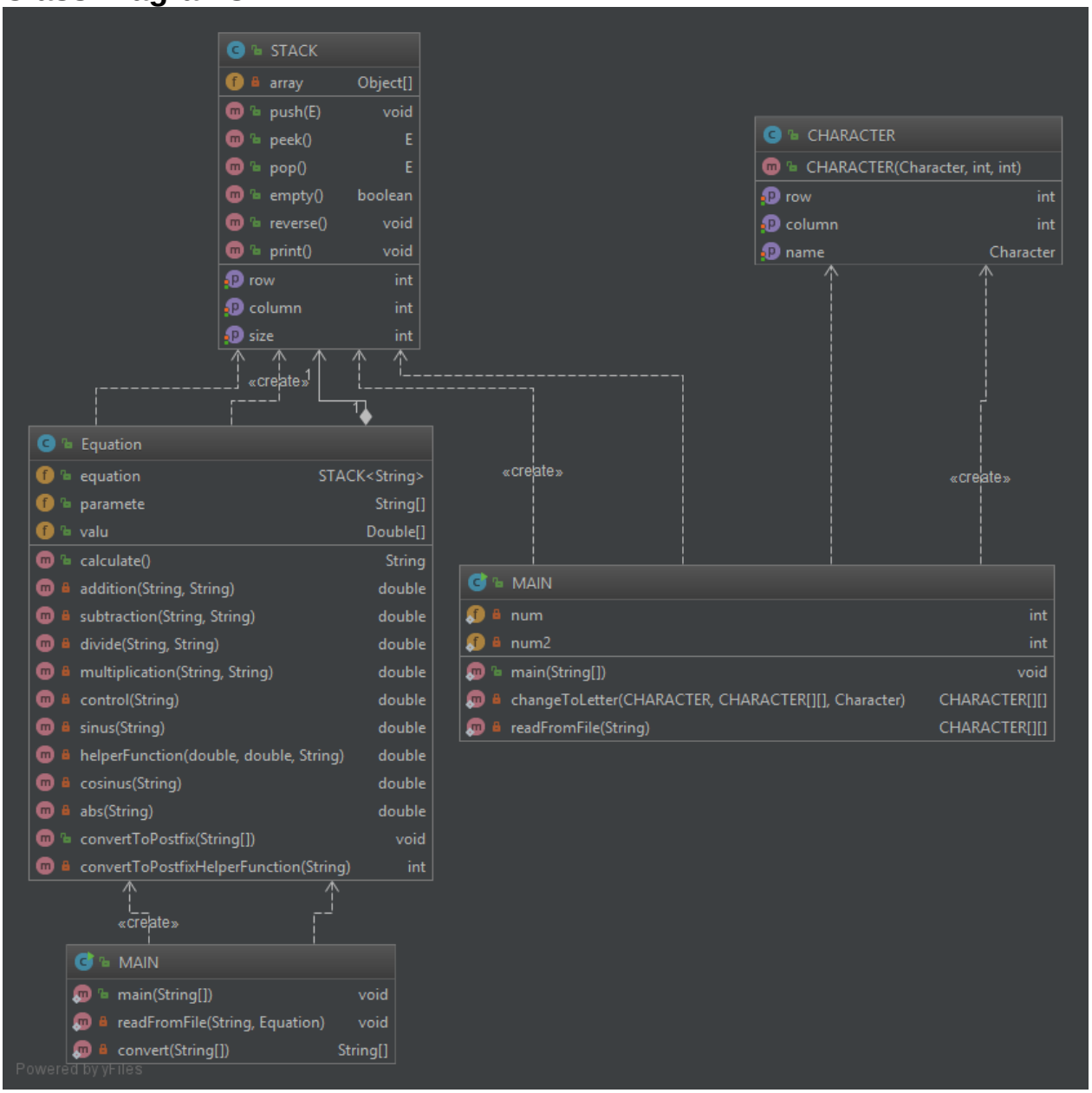
You have to download İntellij IDEA for run the programs. The program requirements are listed below.

Microsoft Windows 10/8/7/Vista/2003/XP (incl.64-bit)
2 GB RAM minimum, 4 GB RAM recommended
1.5 GB hard disk space + at least 1 GB for caches
1024x768 minimum screen resolution

You can download program here;
https://www.jetbrains.com/idea/download/#section=windows (Community Version)

2 METHOD

2.1 Class Diagrams



2.2 Use Case Diagrams

2.2.1 Part1

For the first part, the user must submit a file containing a matrix of 1's and 0's as input to the program. Then you can see how many different groups of numbers in the matrix entered by running the program. The user does not need to do anything other than export input files.

2.2.2 Part2

For the second part, the user has to provide a file with an equation in it. In addition to the parameters used in the equation in the file can write. Then, when he runs the program, he will get the result of his equation as output. The user does not need to do anything other than export input files.

2.3 Problem Solution Approach

2.3.1 Part1

In the first part, I define each element of the matrix I received from the user as the object clas object. Then I keep these objects in the characer type array. Then in the loop, I send each element of the array to the changeToLetter function. In this function, I find the neighbors that are 1 in the list and add them to the stack class object that I created. Then I change all these elements one by one with the next letter. By the end of the cycle, all 1's are converted to letters. Lastly, I print the number of letters in the list.

n = number of elements in the file

```
CHARACTER[][] readFromFile(String filename)
```

This function has two independent cycles. Both of the cycles return n. In total there is a time of 2, which is equal to n bire. T (n) = O (n).

```
CHARACTER[][] changeToLetter(CHARACTER obj, CHARACTER[][] arr,Character character)
```

There is one loop in this function, the loop turns n time. T (n) = O (n).

```
void push(E item)
```

There is one loop in this function, the loop turns n time. T (n) = O (n).

```
E peek()
```

There is no loop in this function. Just return E value. So; T (1) = O (1).

```
E pop()
```

There is one loop in this function, the loop turns n time. T (n) = O (n).

```
void reverse()
```

There is one loop in this function, the loop turns n time. T (n) = O (n).

2.3.2 Part2

In the second part I read the equation and the parameters from the file. I'm throwing them at a strike. Then I edit the equation according to parenthesis and throw it into my own stack. Then I turn the equation in the stack into a postfix state. Then I start to solve the dennklemi process priority. I find the operators in the function and send the parameters to the functions of par + lar, ler - ”,“ * ”,“ / “operators. I check whether the parameters within the functions are one of the parameters in the file or a double number. In addition, I check whether the parameter is cos, sin or abs. I'm going to return the result of the transactions and add them back to the stack until the stack is empty. When the stack is empty, the result of the equation is found and the result is printed on the screen.

```
String calculate()
```

There is one loop in this function, the loop turns n time. T (n) = O (n).

```
double addition(String x, String y)
```

There is no loop in this function. Just return double value. So; T (1) = O (1).

```
double subtraction(String x, String y)
```

There is no loop in this function. Just return double value. So; T (1) = O (1).

```
double divide(String x, String y)
```

There is no loop in this function. Just return double value. So; T (1) = O (1).

```
double multiplication(String x, String y)
```

There is no loop in this function. Just return double value. So; T (1) = O (1).

```
double control(String temp)
```

There is one loop in this function, the loop turns n time. T (n) = O (n).

```
double sinus(String temp)
```

In this function, there are 2 independent loops and these loops are only entered in certain conditions, and when they enter, the loops rotate to the size of the incoming string. It does not turn n time. T (1) = O (1).

```
double cosinus(String temp)
```

In this function, there are 2 independent loops and these loops are only entered in certain conditions, and when they enter, the loops rotate to the size of the incoming string. It does not turn n time. T (1) = O (1).

```
double abs(String temp)
```

In this function, there are 2 independent loops and these loops are only entered in certain conditions, and when they enter, the loops rotate to the size of the incoming string. It does not turn n time. T (1) = O (1).

```
double helperFunction(double a,double b,String c)
```

There is no loop in this function. Just return double value. So; T (1) = O (1).

```
void convertToPostfix(String [] arr)
```

There is one loop in this function, the loop turns n time. T (n) = O (n).

```
int convertToPostfixHelperFunction(String temp)
```

There is no loop in this function. Just return integer value. So; T (1) = O (1).

```
void readFromFile(String filename, Equation eq)
```

There are 3 independent cycles in this function. Each loop rotates by the number of rows in the file.T (1) = O (1).

```
String[] convert(String [] arr)
```

There are 3 independent cycles in this function. He spins the loop. This is 3n.
3n=n T (n) = O (n).

3 RESULT

3.1 Test Cases

3.1.1 Part1

```
0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0
0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0
0 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
```

I tested my program with this input and I check the result and count the white compenents. I think the program works correctly because it is the same as my testimle program.

3.1.2 Part1

```
y=3
z=16

( y + sin( y * z ) ) + ( z * abs( -10.3 ) )
```

I tested my program with this input and I check the result and count the white compenents. I think the program works correctly because it is the same as my testimle program.

3.2 Running Results

3.2.1 Part1

```
168.12540919548218

Process finished with exit code 0
```

3.2.2 Part2

```
"C:\Program Files\Java\jdk-10.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDE
0 0 A A A A A 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 B B B B 0
0 0 0 0 A A A 0 0 0 0 0 0 0 0 0 0 0 0 0 C C C C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 B B B B
0 0 0 0 A A A 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 B B B B
0 0 0 0 A A A 0 D D D 0 0 0 0 0 0 0 0 0 0 0 0 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 B B B B 0 0
0 0 0 0 0 A A A 0 D D D 0 0 0 0 0 0 0 0 0 0 0 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 A A A 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 E E E E E E 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 E E E E 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 F F F 0 0 0 0 0 0 0 0 0 0 E E E E 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 G G 0 0 0 0 0 0 0 0
0 0 0 0 0 0 F F F 0 0 0 0 0 0 0 0 0 0 0 0 E E E E 0 0 0 0 0 0 0 0 0 0 0 0 G G 0 0 0 0 0 0 0 0
0 0 0 0 0 0 F F F 0 0 0 0 0 0 0 0 0 E E E E 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 G 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 E E E E 0 0 0 0 0 0 0 0 0 0 0 0 0 0 G G G 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 H H H 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 H H H 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 I I I I
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 I I I I

Number of white components 9

Process finished with exit code 0
```