**Gebze Technical University**

**Computer Engineering**

**CSE 222 - 2019 Spring**

**HOMEWORK 3 REPORT**

**AHMET MELIH YANALAK**

**151044044**

Course Assistant:

OZGU GOKSU

# INTRODUCTION

## Problem Definition

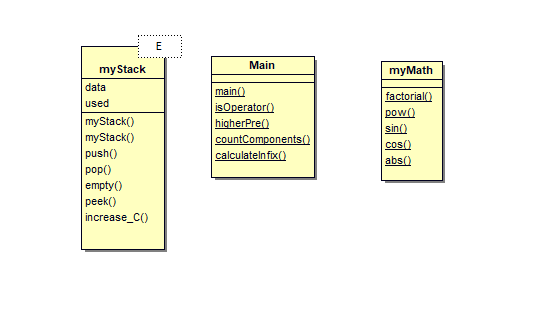
In this Project,there were 2 problems to deal with.First problem is determining the number of components that represented by boolean variables in 2d matrix.Second one is to calculate an infix expression after converting it to postfix expression by using self-implemented stack.

## System Requirements

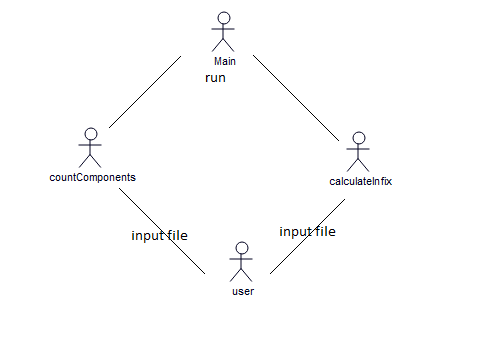
These algorithms require at most 128KB memory,any operating system that can run intelliJ 2018-3.5 app properly and also Java.io library.

# METHOD

## Class Diagrams



## Use Case Diagrams



## Problem Solution Approach

For the first method that counts the number of the components,i used two stacks of integer to keep the indexes of neighbours(with value 1) for 2 dimension.If my current value is 0,then it continues checking the next till it find the value ‘1’.After finding it,then the program also checks its neighbours and add their indexes to stack if their values are 1.Then it pops the stack and mark them as visited(with letter) and continue checking till the reaching end of 2d matrix.The complexity of this method is O(n).(n is number of elements in matrix).Because we have 2 while loop to control all elements in matrix and stack methods are O(1) because of adding to the last and removing the last with using underlying array.

In the second method that calculates the value of an infix expression i used a stack to keep the operators while converting the expression to postfix expression which will lead me to calculate easier later.When the program sees a variable or number it directly copy it to new postfix array.And stack helps to replace the operators at the right position according to precedence of operators.After converting infix to postfix,to calculate the expression i used stack again but this time stack will keep the values of numbers and results as double.When the program sees an operator it pops 2 values if the operator is binary,pops just one value if the operator is unary.If it sees an variable, it replace with its value we read it from file before.After calculating the result we just pop the stack and return its value.The complexity of this program is O(n^2) because of the control whether it reaches the empty space or not i needed while loop inside main loop.

# RESULT

## Test Cases

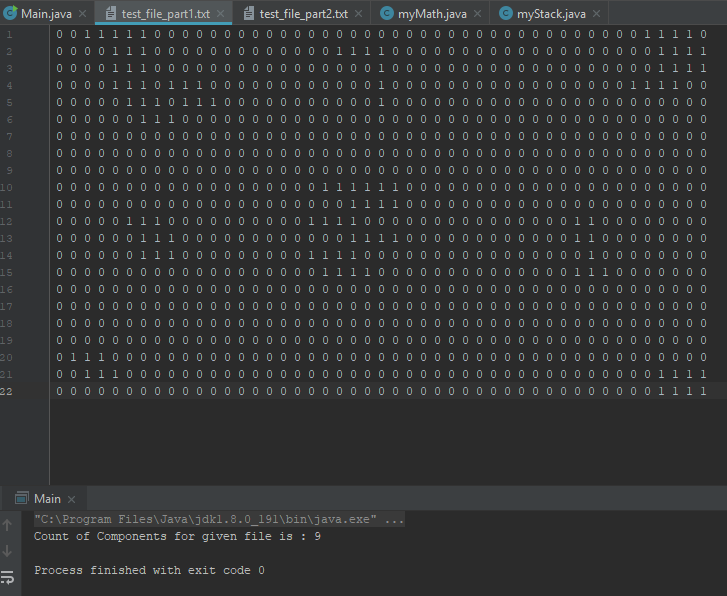
For the first part i tested the possible worst case which is an input file with full of 1 values

And the best case which is an matrix with all 0 elements.And i also tried the given test file

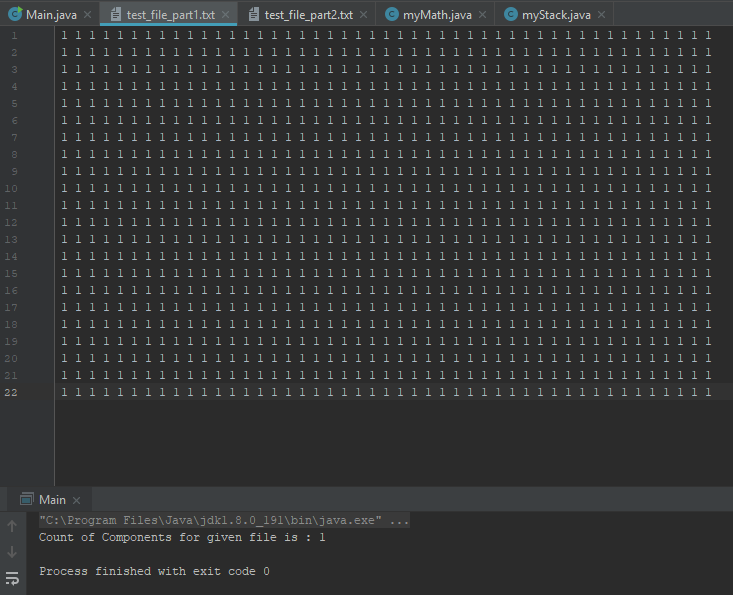
To control.

For the second method i tested all the operators that can be used in expression.that i implemented myself (sin cos abs),operators that are binary(+,-,\*,/) and also unary (+,-).I also tested with negative variables and negative numbers.

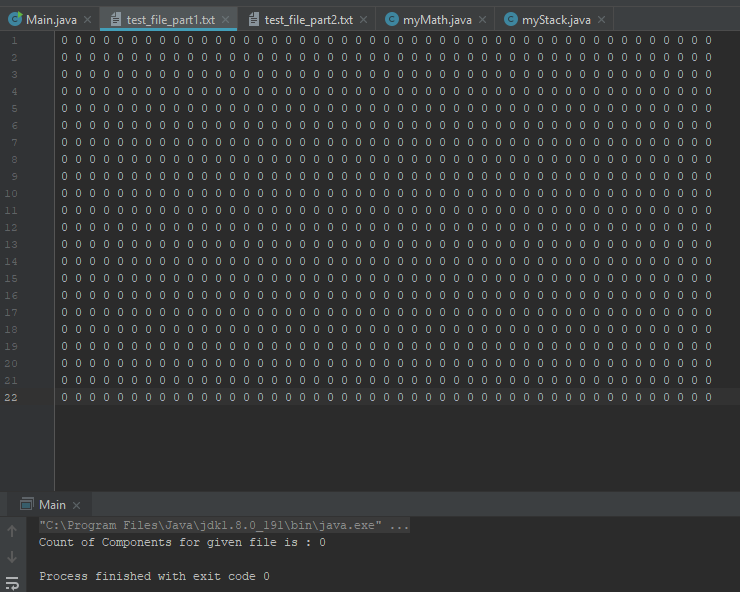
## Running Results



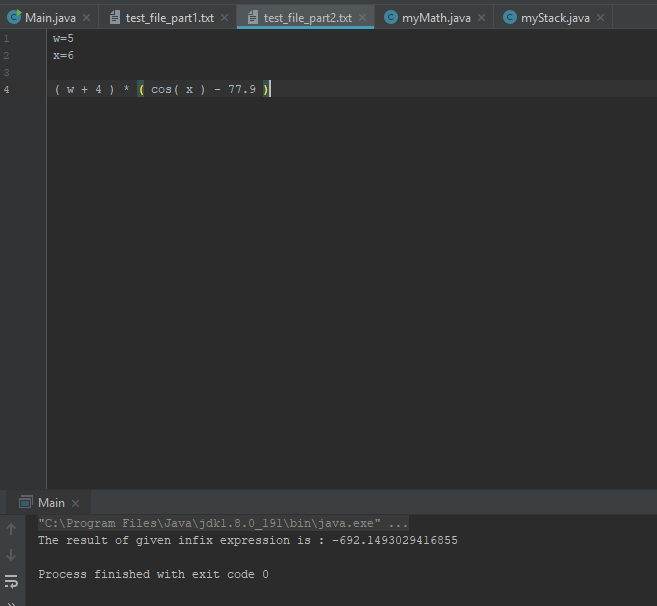
Result with input file given through moodle page



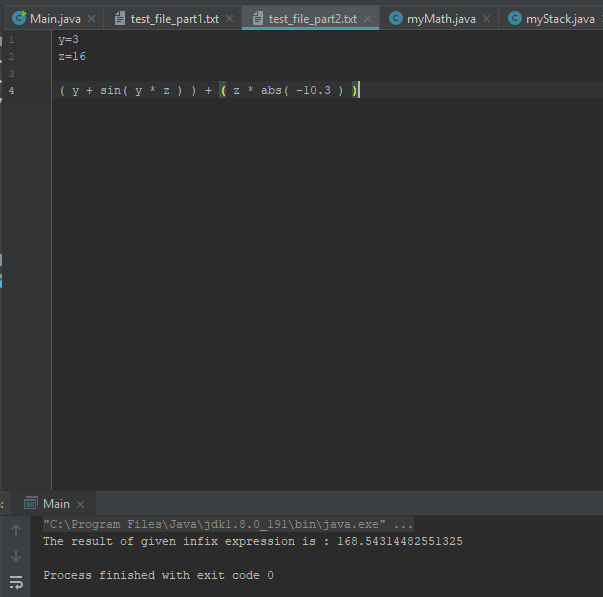
Result with all-1 matrix input



Result with file input all-0 matrix



Result for the input file that given through moodle



Result for another input file given through moodle