Gebze Technical University
Computer Engineering

CSE 222 - 2019 Spring

HOMEWORK 3 REPORT

Taylan ÖNDER
151044015

Course Assistant: Özgü GÖKSU

1 INTRODUCTION

1.1 Problem Definition

Part 1

In the first part the problem is calculate components number in matrix array.In this part our time complexity is important.So the requested solution from us is approximately O(n).Matrix will be read in a file and according the ‘1’ number decide how many components are there in matrix.

Part 2

In the second part the porblem is to be able to calculate the result by converting a given statement to postfix. Variables and expressions need to be read in the given file .Using the stack structure variables,functions (sin,cos,abs) and paranthesis solved with group and calculate by placing the values of the variables.

1.2 System Requirements

Part 1:

We need MatrixStack (my stack class) object to keep index of array which include components elements. We need two dimensional char array to keep matrix of integers..File should read twice.Matrix class has inner node class to keep index of components x and y.This program work in terminal and show result.

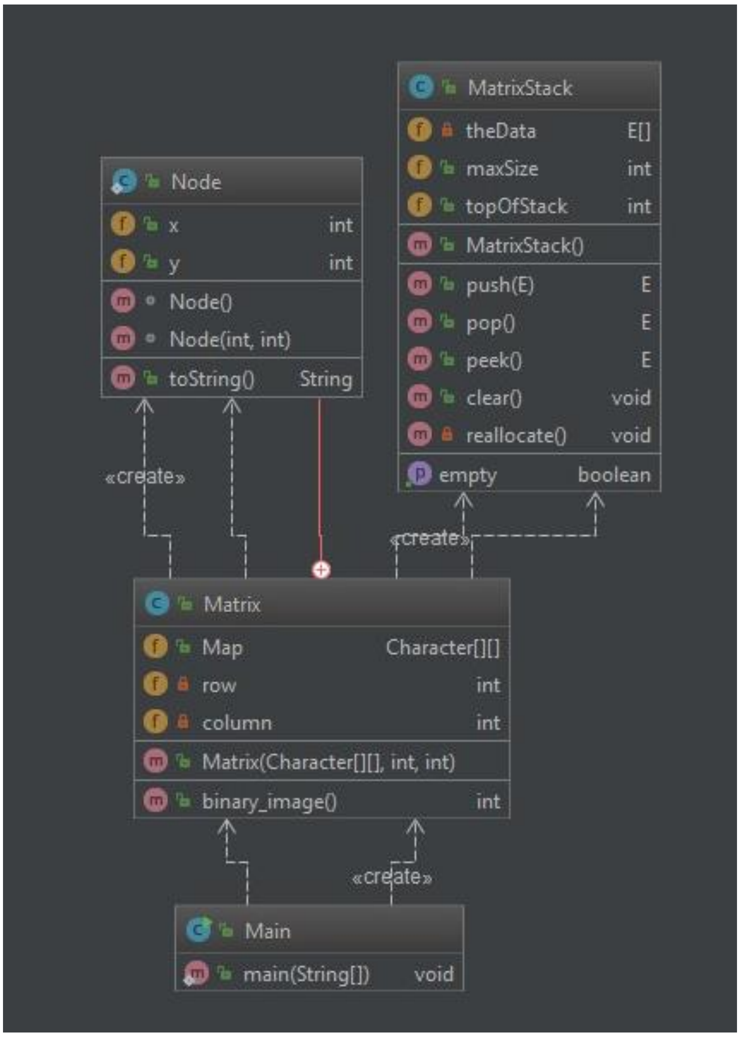
Part 2

We need stackPostfix (my stack class) object to keep operators and paranthesis,Calculate class object need to change postfix status and calculate the according to given variables and variable values. This program work in terminal and show result.

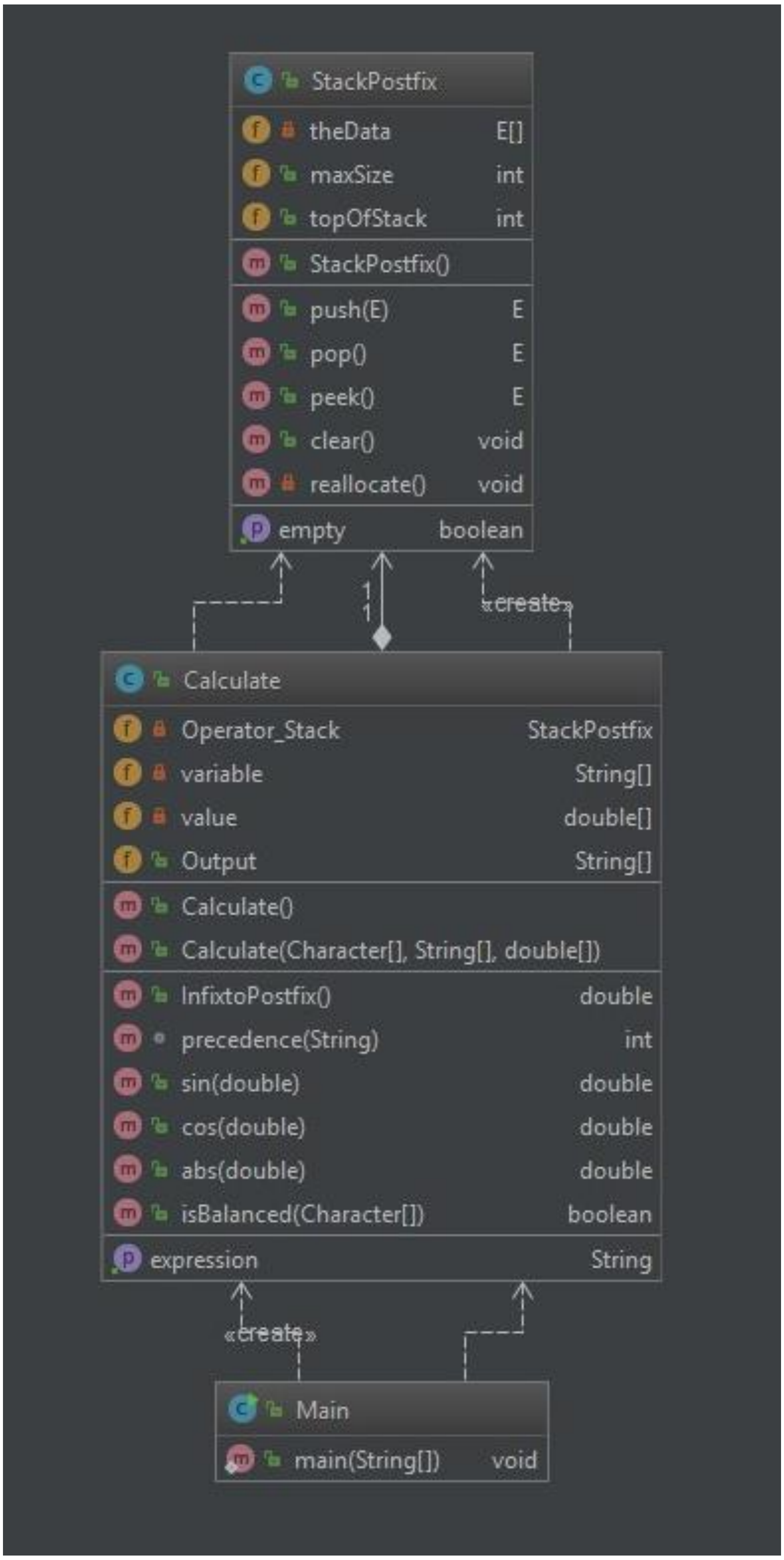
2 METHOD

2.1 Class Diagrams

Part 1

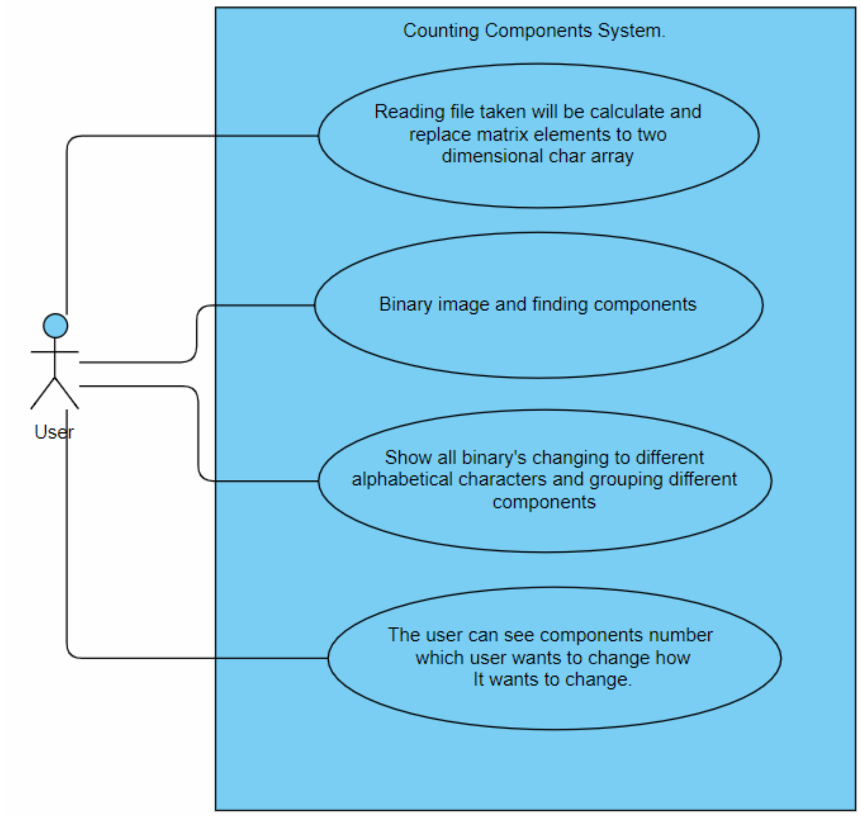


Part 2

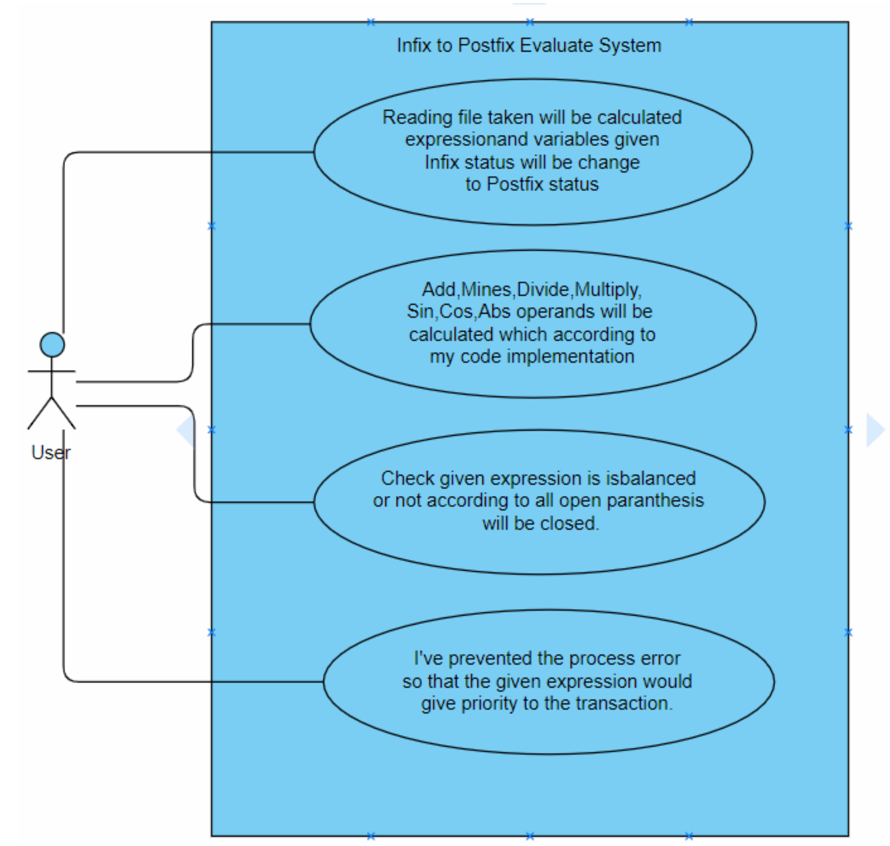


2.2 Use Case Diagram

Part 1


Counting Components System.
Reading file taken will be calculate and replace matrix elements to two dimensional char array
Binary image and finding components
Show all binary's changing to different alphabetical characters and grouping different components
The user can see components number which user wants to change how It wants to change.
User

Part 2


Infix to Postfix Evaluate System
Reading file taken will be calculated expressionand variables given Infix status will be change to Postfix status
Add,Mines,Divide,Multiply, Sin,Cos,Abs operands will be calculated which according to my code implementation
Check given expression is isbalanced or not according to all open paranthesis will be closed.
I've prevented the process error so that the given expression would give priority to the transaction.
User

2.2 Problem Solution Approach

Part 1

First of all file should read twice. We need two dimensional char array to keep matrix of integers.So first reading we find array of column and raw after allocate two dimensional char array.Second reading matrix elements of integer will be replace to char array.After that create MatrixStack (my stack class) object to keep index of array which include components elements.I created matrix class object to copy all matrix elements and find components.I implement binary_image method to look all array of element and find components.I checked left,right,up and down index and if include ‘1’ push in stack.When the neightbours doesn’t include ‘1’ all stack element pop and I return first starting index and check all index array elements.And calculate all different components and keep counter all different components.When components change I changed array element to different alphabetical characters to show components.

Part 2

First of all I decide to solve problem using stack and I implement stack class with template array.I implement one class call calculate to change infix status to postfix. Calculate class constructure take reading file expressions variables and variables values.Infixtopostfix method change infix to postfix using stack and string array and calculate according to given variable and values.Precedens method is a helpful method to pop stack elements according to precedence of methods.I implement helpfully methods Sinus,cosinus and abs method for don’t use math library.Isbalanced method for check paranthesis open and close.

3 RESULT

3.1 Test Cases

Part 1

1. Firstly reading file and find row and column of two dimensional char array.(Example given txt file in moodle.)
2. Secondly reading file to replace integers allocate char array.
3. Thirdly create matrix object and copy all data in object fields.
4. Fourtly call binary_image function to calculate components number and change array elements.
5. Print changing array and components number.
6. All test cases show below results.

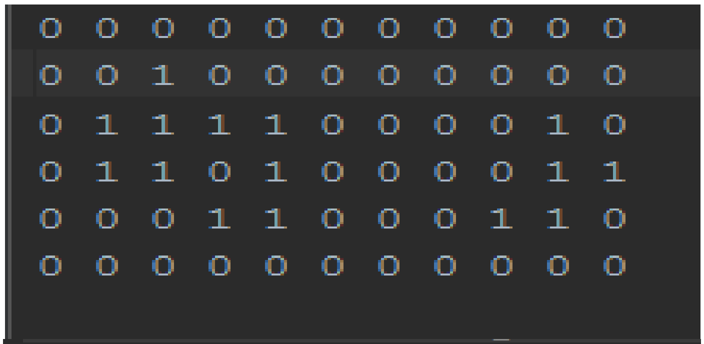
Part 2

1. Reading file and expression is (w + 4) * (cos(x) - 77.9) , variables are w and x.Variable values are w=5, x=6.
2. Check balanced to call isbalanced method and throw exception or not.

3. Create Calculate class object to copy of expression and variables name and variables values.
4. Call Calculate InfixtoPostfix method and change ınfix status to postfix and calculate result of expression.
5. Print screen result of expression.
6. All test cases show below results.

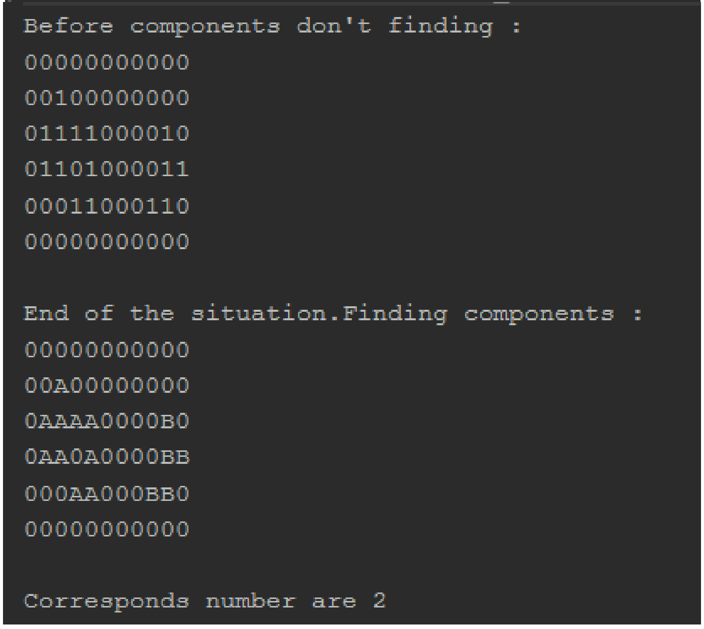
3.2 Running Results

Part 1



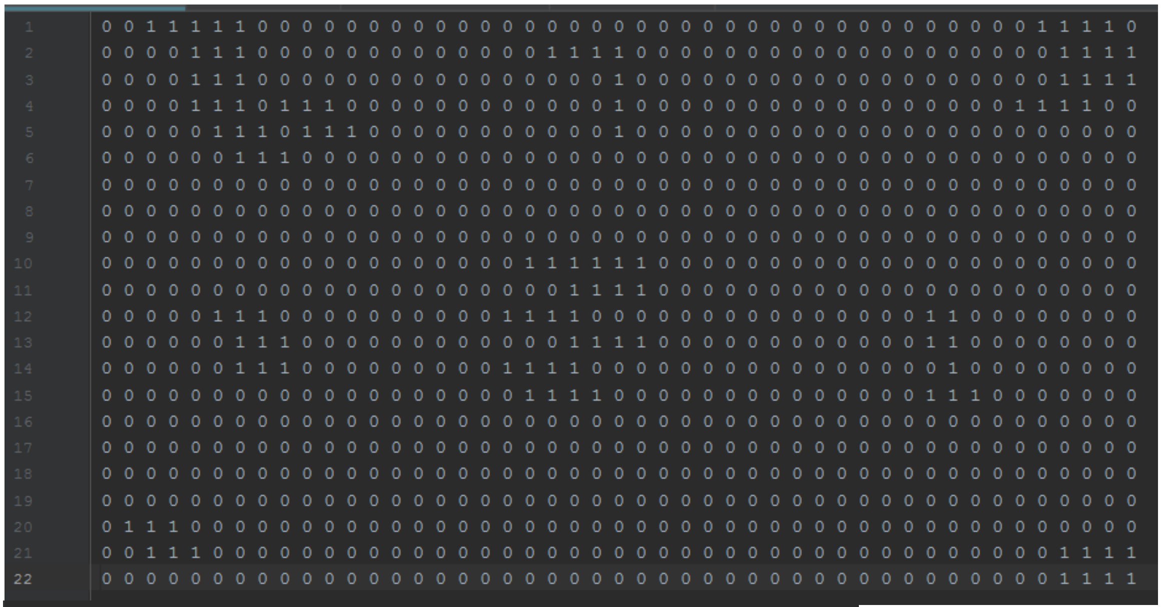
```
0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0
0 1 1 1 1 0 0 0 0 1 0
0 1 1 0 1 0 0 0 0 1 1
0 0 0 1 1 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 0
```

Input



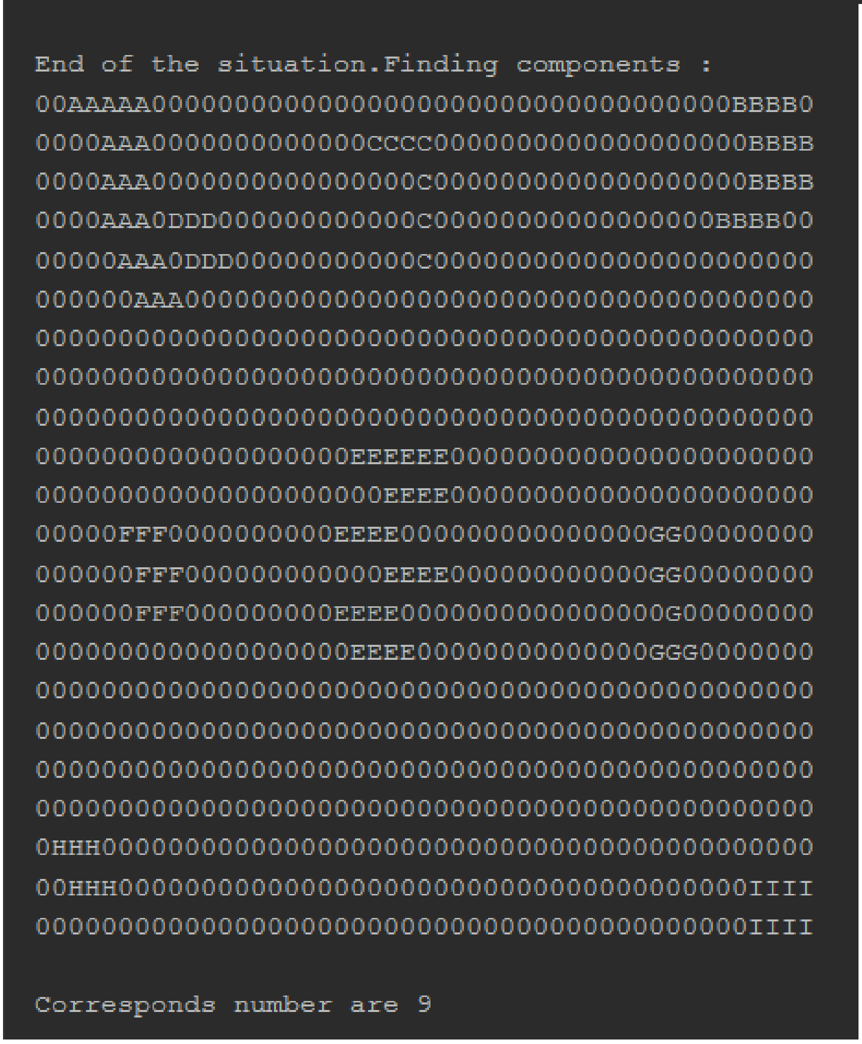
```
Before components don't finding :
00000000000
00100000000
01111000010
01101000011
00011000110
00000000000

End of the situation.Finding components :
00000000000
00A00000000
0AAAA0000B0
0AA0A0000BB
000AA000BB0
00000000000

Corresponds number are 2
```

Output



```
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
```

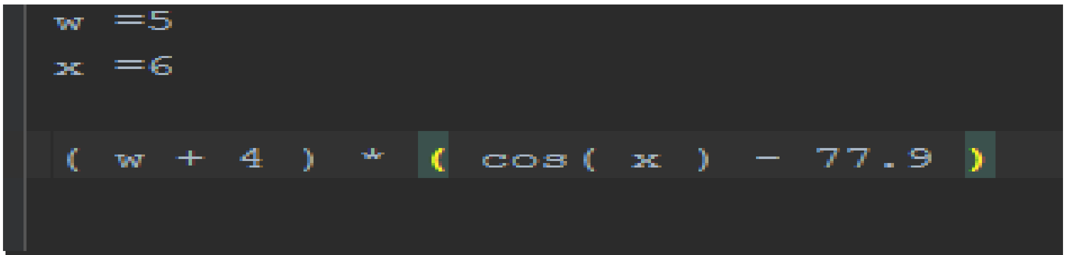
Input



```
End of the situation.Finding components :
00AAAAA00000000000000000000000000000000000BBBB0
0000AAA0000000000000CCCC0000000000000000000BBBB
0000AAA0000000000000000C0000000000000000000BBBB
0000AAA0DDD000000000000C00000000000000000BBBB00
00000AAA0DDD00000000000C00000000000000000000000
000000AAA00000000000000000000000000000000000000
00000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000
0000000000000000000EEEEEE0000000000000000000000
000000000000000000000EEEE0000000000000000000000
00000FFF0000000000EEEE000000000000000GG00000000
000000FFF000000000000EEEE000000000000GG00000000
000000FFF000000000EEEE0000000000000000G00000000
0000000000000000000EEEE00000000000000GGG0000000
00000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000
0HHH0000000000000000000000000000000000000000000
00HHH00000000000000000000000000000000000000IIII
0000000000000000000000000000000000000000000IIII

Corresponds number are 9
```

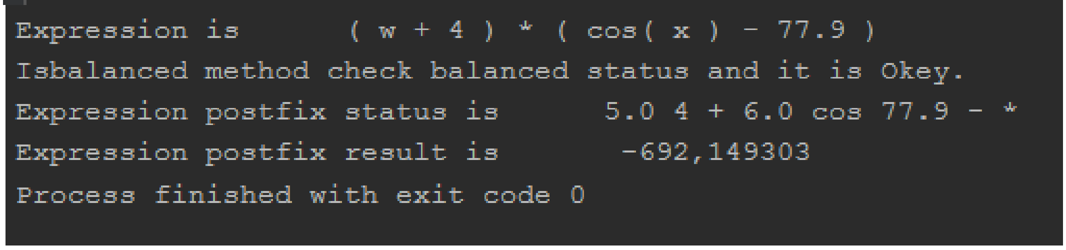
Output

Part 2



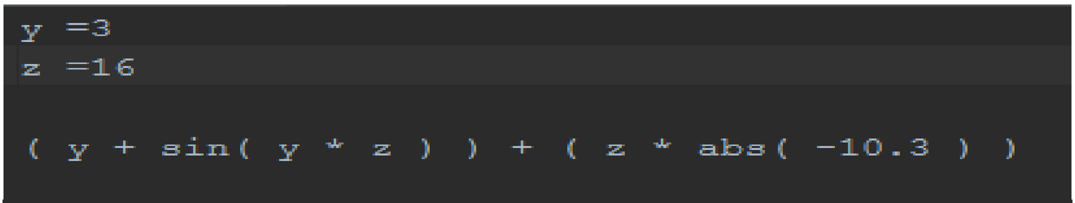
```
w =5
x =6

( w + 4 ) * ( cos( x ) - 77.9 )
```

Input



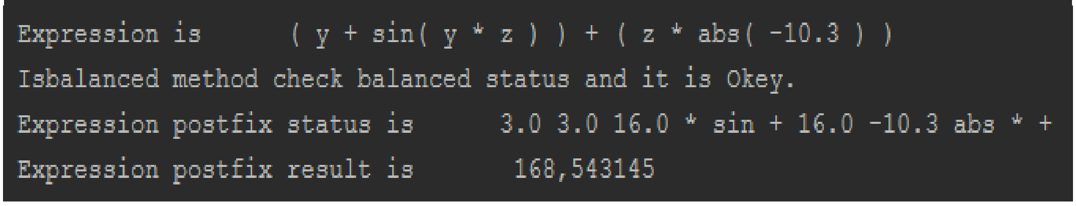
```
Expression is       ( w + 4 ) * ( cos( x ) - 77.9 )
Isbalanced method check balanced status and it is Okey.
Expression postfix status is       5.0 4 + 6.0 cos 77.9 - *
Expression postfix result is        -692,149303
Process finished with exit code 0
```

Output



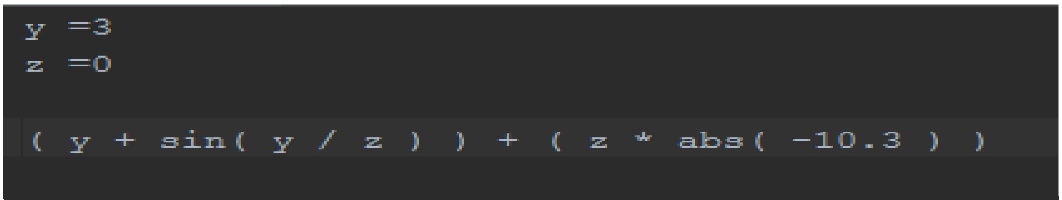
```
y =3
z =16

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

Input



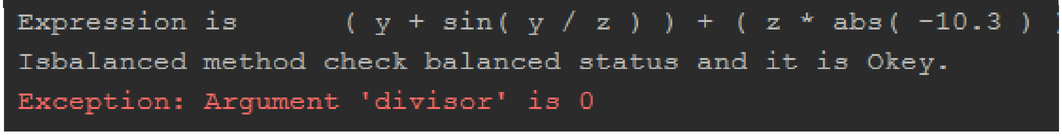
```
Expression is       ( y + sin( y * z ) ) + ( z * abs( -10.3 ) )
Isbalanced method check balanced status and it is Okey.
Expression postfix status is      3.0 3.0 16.0 * sin + 16.0 -10.3 abs * +
Expression postfix result is       168,543145
```

Output



```
y =3
z =0

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

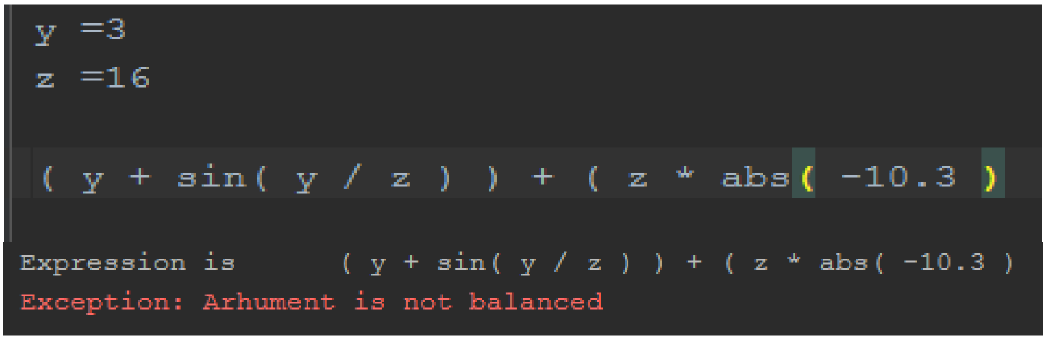
Input



```
Expression is       ( y + sin( y / z ) ) + ( z * abs( -10.3 )
Isbalanced method check balanced status and it is Okey.
Exception: Argument 'divisor' is 0
```

Output

Check divide 0 and throw exception (IllegalArgumentException).



```
y =3
z =16

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

Expression is      ( y + sin( y / z ) ) + ( z * abs( -10.3 )
Exception: Arhument is not balanced
```

Input

Output

1 brackets missing and throw exception (EmptyStackException).

4. Time Complexity

Part1

Methods	Time Complexity	
Matrix Class		
Matrix()	O(n)	
binary_image()	O(m*n)	
Matrix(Character[][]matrix_map,int row_map,int column_map)	O(n)	
MatrixStack Class		
MatrixStack	O(1)	
Push()	O(1)	
Pop()	O(1)	
Peek()	O(1)	
Clear()	O(m)	
Isempty()	O(1)	
Reaalocate()	O(m)	

Part1 of time complexity is O(m*n).Worst case o(m*n), Average case O(m*n), Best case O(n).

Methods	
Calculate Class	
Calculate()	O(1)
getExpression()	O(1)
Calculate(Character []first_expression,String []data_variable,double []value_of_variable)	O(3*n)
InfixtoPostfix	O(n*n+3*n)
Precedence()	O(1)
sin	O(m)
Cos	O(m)
abs	O(1)
isBalanced(Character[] first_expression)	O(2*n)

Part2 of time complexity is O(n*n).Worst case o(n*n+3*n), Average case O(n*n+3*n), Best case O(n*n).