**Lab report Question 1**

Write a program that simulates a calculator. It reads three integers and two characters. X Multiplication of two numbers D Quotient of two numbers M Remainder of two numbers A Addition of two numbers S Subtraction of two numbers The sequence of operand should follow this order X D M A S.

**Problem Description on the problem**

The question is we are going to make a program that simulates a simple calculator. This program will started by read three integers and two operands. By then the program will run based on the operand.

X is Multiplication of two numbers.

D is Quotient of two numbers.

M is Remainder of two numbers.

A is Addition of two numbers.

S is Subtraction of two numbers.

In Mathematics we do multiplication, quotient and reminder first then come to addition and subtraction. Therefore a problem in this question is we need to consider which operand input by user and arrange the priority of the operands in order to calculate the correct output.

**Solution**

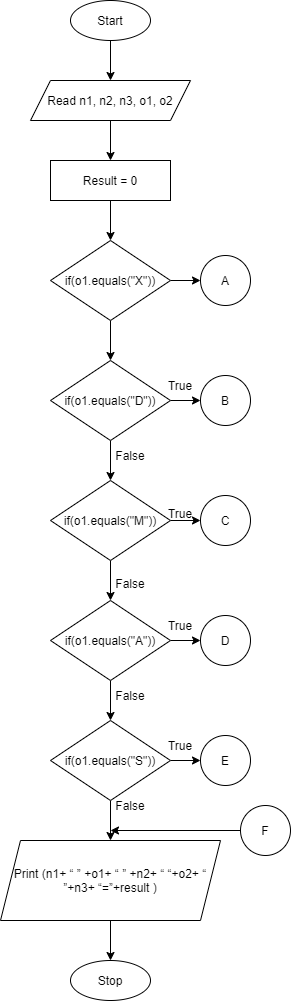
In writing code, we need to use nested if statements to determine the first and second operands, by then we make calculation for each and every pair of operands. For example if user input E F G as three integers and operand 1 is A(addition) and operand 2 is X(multiplication), we need to use an if statement to catch the first operand A and inside this if statement there are another if statements that catch second operand X. Then the system can arrange operand X come before operand A. Result, E + (F\*G).

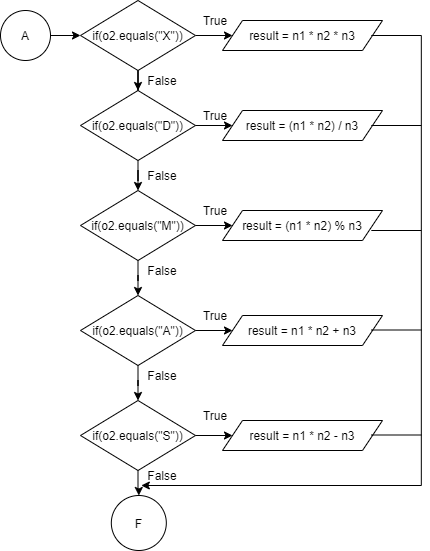
**Pseudocode**

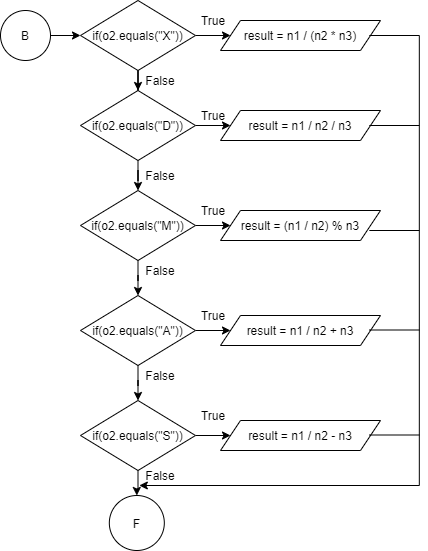
Input: n1, n2, n3, o1, o2

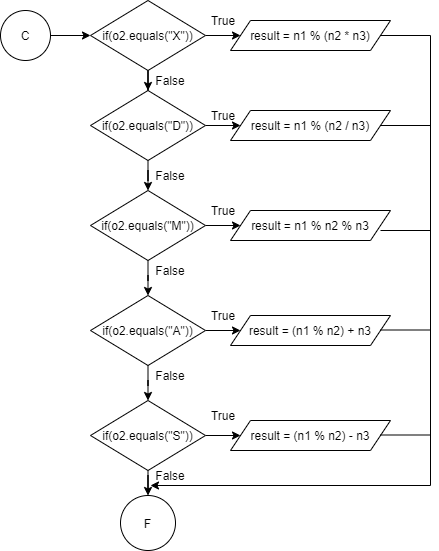
Process: Calculate and display result based on n1, n2, n3, o1, o2

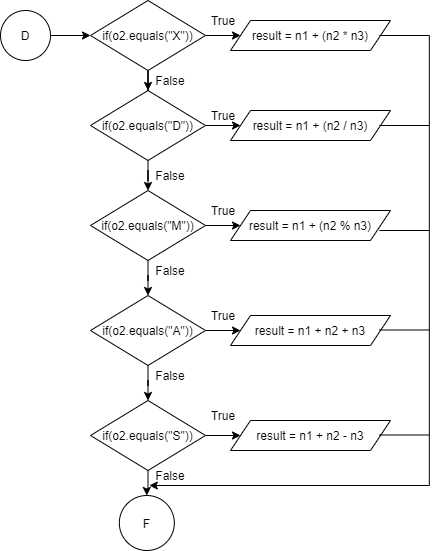
Output: result

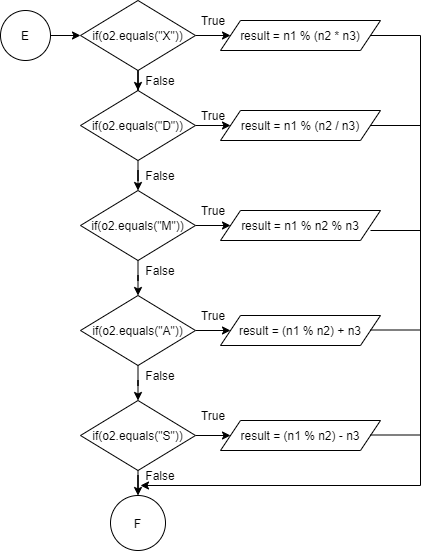
**Flowchart**



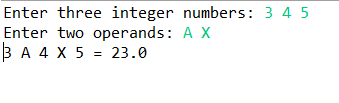


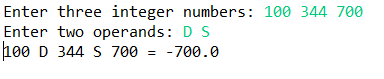
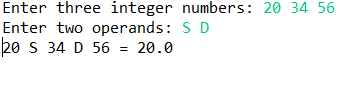




****

**Sample Input and Output**







**Source Code**

import java.util.Scanner;  
public class JavaApplication1 {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.in);  
  
 // user input the integer and operand  
 System.out.print("Enter three integer number: ");  
 String numberInput = sc.nextLine();  
 System.out.print("Enter two operands: ");  
 String operandInput = sc.nextLine();  
  
 // process the user input  
 String[] numStr = numberInput.split(" ", 3);  
 String[] ope = operandInput.split(" ", 2);  
  
 int n1 = Integer.parseInt(numStr[0]);  
 int n2 = Integer.parseInt(numStr[1]);  
 int n3 = Integer.parseInt(numStr[2]);  
 String o1 = ope[0];  
 String o2 = ope[1];  
 double result = 0;  
  
 // conduct the calculation according to the precedence  
 // X D M A S in order  
 if (o1.equals("X")) {  
 if (o2.equals("X")) {  
 result = (n1 \* n2) \* n3;  
 }  
 if (o2.equals("D")) {  
 result = (n1 \* n2) / n3;  
 }  
 if (o2.equals("M")) {  
 result = (n1 \* n2) % n3;  
 }  
 if (o2.equals("A")) {  
 result = (n1 \* n2) + n3;  
 }  
 if (o2.equals("S")) {  
 result = (n1 \* n2) - n3;  
 }  
 }  
  
 if (o1.equals("D")) {  
 if (o2.equals("X")) {  
 result = n1 / (n2 \* n3);  
 }  
 if (o2.equals("D")) {  
 result = (n1 / n2) / n3;  
 }  
 if (o2.equals("M")) {  
 result = (n1 / n2) % n3;  
 }  
 if (o2.equals("A")) {  
 result = (n1 / n2) + n3;  
 }  
 if (o2.equals("S")) {  
 result = (n1 / n2) - n3;  
 }  
 }  
  
 if (o1.equals("M")) {  
 if (o2.equals("X")) {  
 result = n1 % (n2 \* n3);  
 }  
 if (o2.equals("D")) {  
 result = n1 % (n2 / n3);  
 }  
 if (o2.equals("M")) {  
 result = (n1 % n2) % n3;  
 }  
 if (o2.equals("A")) {  
 result = (n1 % n2) + n3;  
 }  
 if (o2.equals("S")) {  
 result = (n1 % n2) - n3;  
 }  
 }  
  
 if (o1.equals("A")) {  
 if (o2.equals("X")) {  
 result = n1 + (n2 \* n3);  
 }  
 if (o2.equals("D")) {  
 result = n1 + (n2 / n3);  
 }  
 if (o2.equals("M")) {  
 result = n1 + (n2 % n3);  
 }  
 if (o2.equals("A")) {  
 result = (n1 + n2) + n3;  
 }  
 if (o2.equals("S")) {  
 result = (n1 + n2) - n3;  
 }  
 }  
  
 if (o1.equals("S")) {  
 if (o2.equals("X")) {  
 result = n1 - (n2 \* n3);  
 }  
 if (o2.equals("D")) {  
 result = n1 - (n2 / n3);  
 }  
 if (o2.equals("M")) {  
 result = n1 - (n2 % n3);  
 }  
 if (o2.equals("A")) {  
 result = n1 - (n2 + n3);  
 }  
 if (o2.equals("S")) {  
 result = (n1 - n2) - n3;  
 }  
 }  
 // Print out the result of the calculation  
 System.out.println(n1 + " " + o1 + " " + n2 + " " + o2 + " " + n3 + " = " + result);  
 }  
  
}

**Lab Report Question 2**

Write a simple two players dice game. Each player will roll the dice twice and the player with the highest score wins the game. You need to follow the special rules as stated below.

|  |  |
| --- | --- |
| **Rules** | **Score** |
| If first and second dice value are 6 | No score but can roll the dice twice. |
| If first and second dice value are 1 | No score but multiply the existing score by 2. |
| If first and second dice value are both odd number and not both 1. | No score and -5 from the existing score |

1. **Problem Description**

The question requests to create a multi round two-player dice game, each player will roll the dice twice each round and the player with the highest score wins. In this game, the players will determine and input how many rounds their game will be. the score of each round will equals to the sum of value of dice which the players diced.

However, there are special rules for the game. After the player roll the two dices, the program will need to check if the outcome of the two dice equivalent to any of the specific outcome, which are both dice value are 6, both dice value are 1, and both dice value are odd and are not 1.

After all rounds have completed, the program will compare final cumulative score of both player, and displays the player with higher score as the winner of the game.

1. **Solution**

In the program, we create the essential variables: two player’s accumulative score, two random numbers(randomly generated using random method) as the two dice, and the round of the game. Before the game starts, user will be prompted to input total rounds they will play for their gamer. In the game, two dices’ outcome can be obtained by generating two random integers between 1 - 6. A while-loop will be created to repeat the process, with ‘round of the game’ be the counter, the condition controlling the loop. The loop will break after all rounds are completed.

After the game starts and the dices are rolled, by using if-else statements, the program will check if the two dice is equivalent to the specific outcome:

1. If both dice rolled are 6, then the loop will ‘continue’, which skips all the following statements and counter update, re-execute the current round and re-roll the dice.
2. If both dice rolled are 1, then the program will not add the dices’ value to the player’s cumulative score, but instead double the current cumulative score of the player.
3. If both dice rolled are odd number and are not 1, the program will not add the dices’ value to the player’s cumulative score, but instead deduct 5 marks from the player’s cumulative score.

When the while-loop reach the total round of the game and had completed all rounds, the program will then output both players’ cumulative score. Using if-else statements to compare both players’ cumulative score, the program will output the player with the higher score as the player. If both players have the same cumulative score, program will output that the game is a draw.

**Pseudocode**

start

declare int scorePlayer1, scorePlayer2, dice1, dice2, playerTurn, round

initialize scorePlayer1 = 0, scorePlayer2 = 0, playerTurn = 1

Request user to input rounds to play

Store number of rounds in round

while(true)

dice1 = random number between 1-6

dice2 = random number between 1-6

if(playerTurn is odd number)

output “Player 1’s turn” and dice1 and dice2

if(dice1 and dice 2 == 6)

output “Player1’s both dice value are 6, roll again

skips all following statements and re-roll the dice

else if(dice1 and dice2 == 1)

scoreplayer1 = scorePlayer1 \* 2

else if(dice1 and dice2 are odd and are not 1)

scorePlayer1 = scorePlayer1 – 5

else

scorePlayer1 = dice1 + dice2

end if

else if(playerTurn is even number)

output “Player 2’s turn” and dice1 and dice2

if(dice1 and dice 2 == 6)

output “Player2’s both dice value are 6, roll again”

skips all following statements and re-roll the dice

else if(dice1 and dice2 == 1)

scoreplayer2 = scorePlayer2 \* 2

else if(dice1 and dice2 are odd and are not 1)

scorePlayer2 = scorePlayer2 – 5

else

scorePlayer2 = dice1 + dice2

end if

if(playerTurn = round \* 2)

break the while-loop

end if

end while

output scorePlayer1, scorePlayer2

If(scorePlayer1 > scorePLayer2)

output “Player 1 wins!”

else if (scorePlayer1 < scorePLayer2)

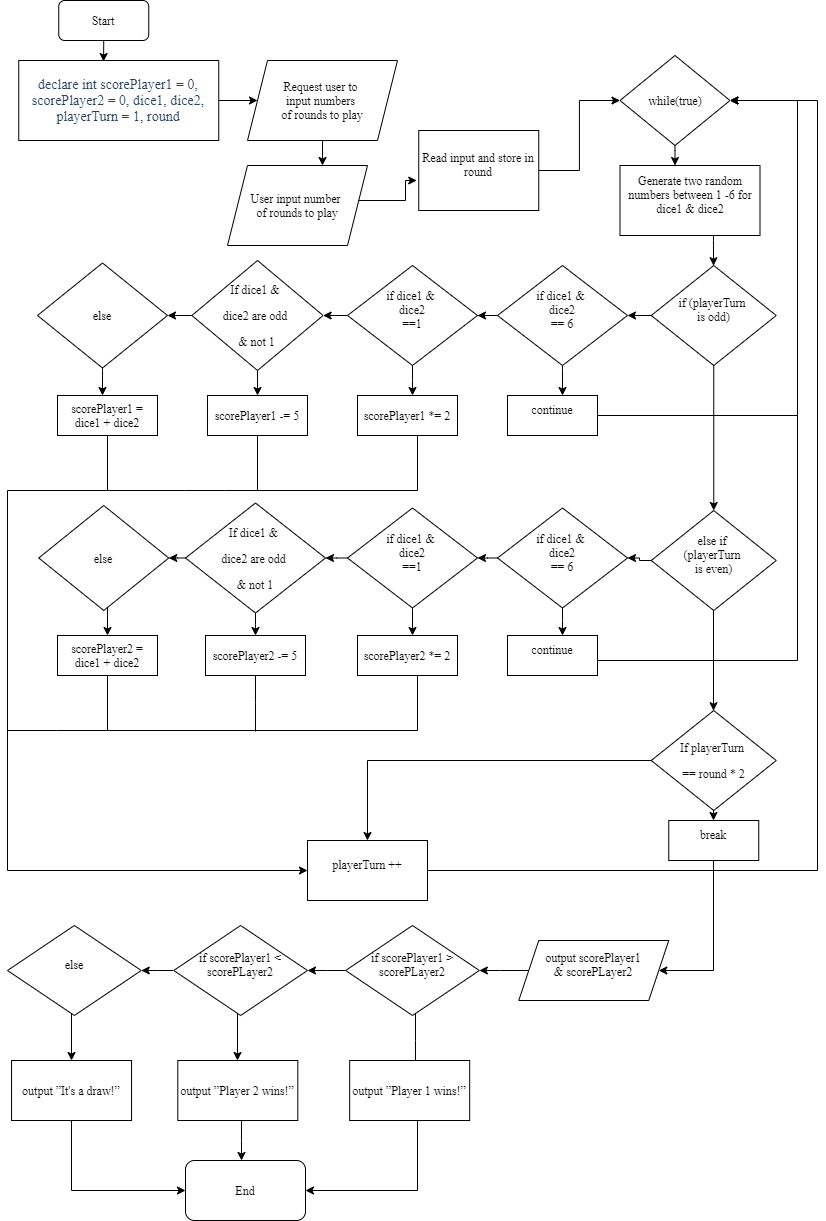
output “Player 2 wins!”

else

output “It’s a draw!”

end if

stop

**Flowchart**

**Sample Input and Output**

**Source Code**

**Question 3**

Cramer’s rule is used to solve the linear equations. ax+by=e, cx+dy=f; x = (ed-bf)/(ad-bc) y=(af-ec)/(ad-bc) Write a program that ask the user to enter two input line in the format of ax+by=e and cx+dy=f. Display the result of x and y. If ad – bc is equal to 0. Display "The equation has no solution". Given the value of a, b, c, d is (1-9)

1. **Problem**

The question requires us to write a program that is able to calculate Cramer’s Rule with the formula:

,

;

while the user inputs two input lines with the formula ax+by=e and cx+dy=f. When ad-bc=0 this particular program should print the line “The equation has no solution” and the program shall limit the range of input variables a,b,c,d to be between integers 1-9 only.

The first requirement of this program would be its ability to extract the coefficients a,b,c,d and equation value e,f from the input lines of user.

The second requirement is to let the original data type of input which is String to be converted into int in order to perform calculation.

The third requirement is to print out messages indicating false input or no solution for that equation.

1. **Solution**

To extract the coefficient in the input lines of user, we will need to use the method ‘String split()’ to perform this task. We indicate the parts where we want to separate from the input with the use of characters specified in an array. Since we will need to separate a,b,c,d,e,f from ax+by=e, cx+dy=f, the set used to identify where to split will be ‘[x+y=]+’. Resulting of this String split, we will have two String arrays (one from each input) with each 3 elements. [ ] means that it will check x or + or y or =, and the ‘+’ outside of the [] means that it will remove the blank spaces in the array

In order to perform calculation using the extracted String array elements, we will need to convert the data type of the array elements from String to int. To perform this, we will use the method ‘Integer.parseInt()’ to convert each elements of the arrays to int.

To check if the input of the user is applicable to the Cramer’s Rule formula, we will use ‘if’ to screen the input before performing the calculation. If the input is not applicable, an error message will be printed out to the user.

**Pseudocode:**

user input equation 1

read equation 1

initialise input 1 = “(equation 1)”

user input equation 2

read equation 2

initialise input 2= ”(equation 2)”

split input 1, input 2 to get coefficient of x, y and the equation value

parse String to int

declare variables to a,b,c,d,e,f

limit variables from 1-9

if ( a <-9 || b <-9 || c<-9 || d<-9 || a>9 || b>9|| c>9|| d>9) {

print“ The input is invalid”

}

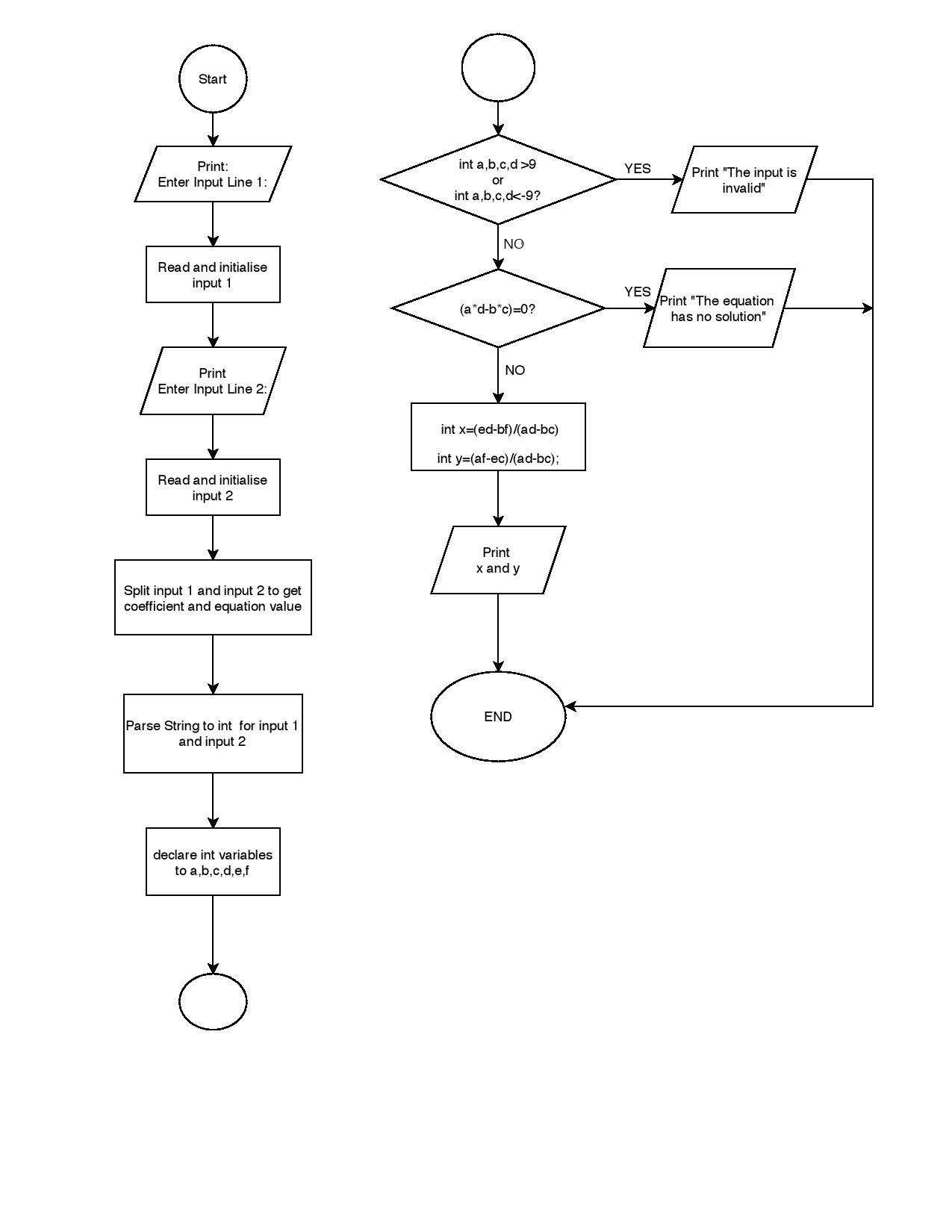
if (ad-bc=0) print “The equation has no solutions”

else {

int x= (ed-bf)/(ad-bc);

int y= (af-ec)/(ad-bc);

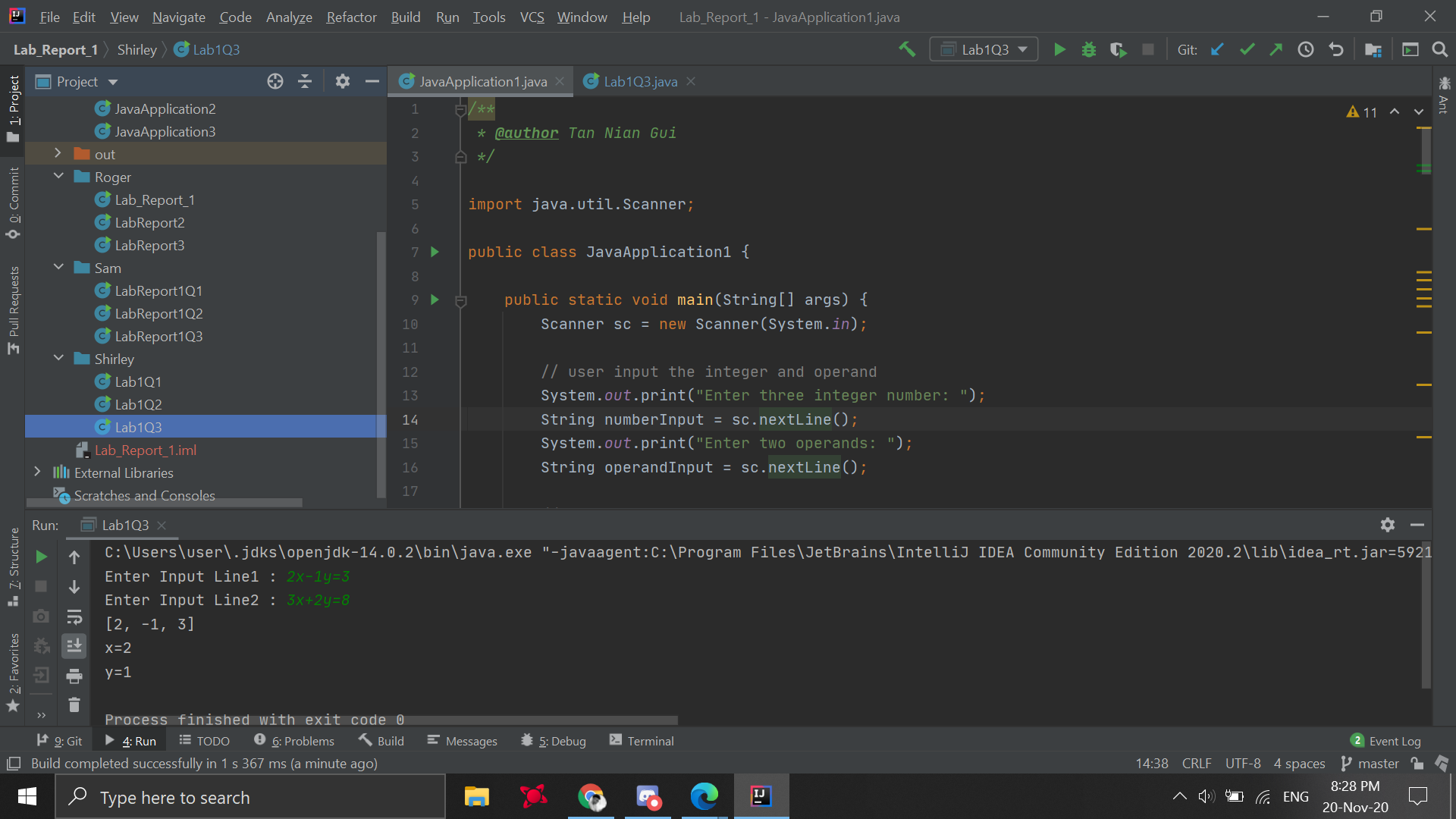
}

**Flowchart:** 

**Sample Input and Output**

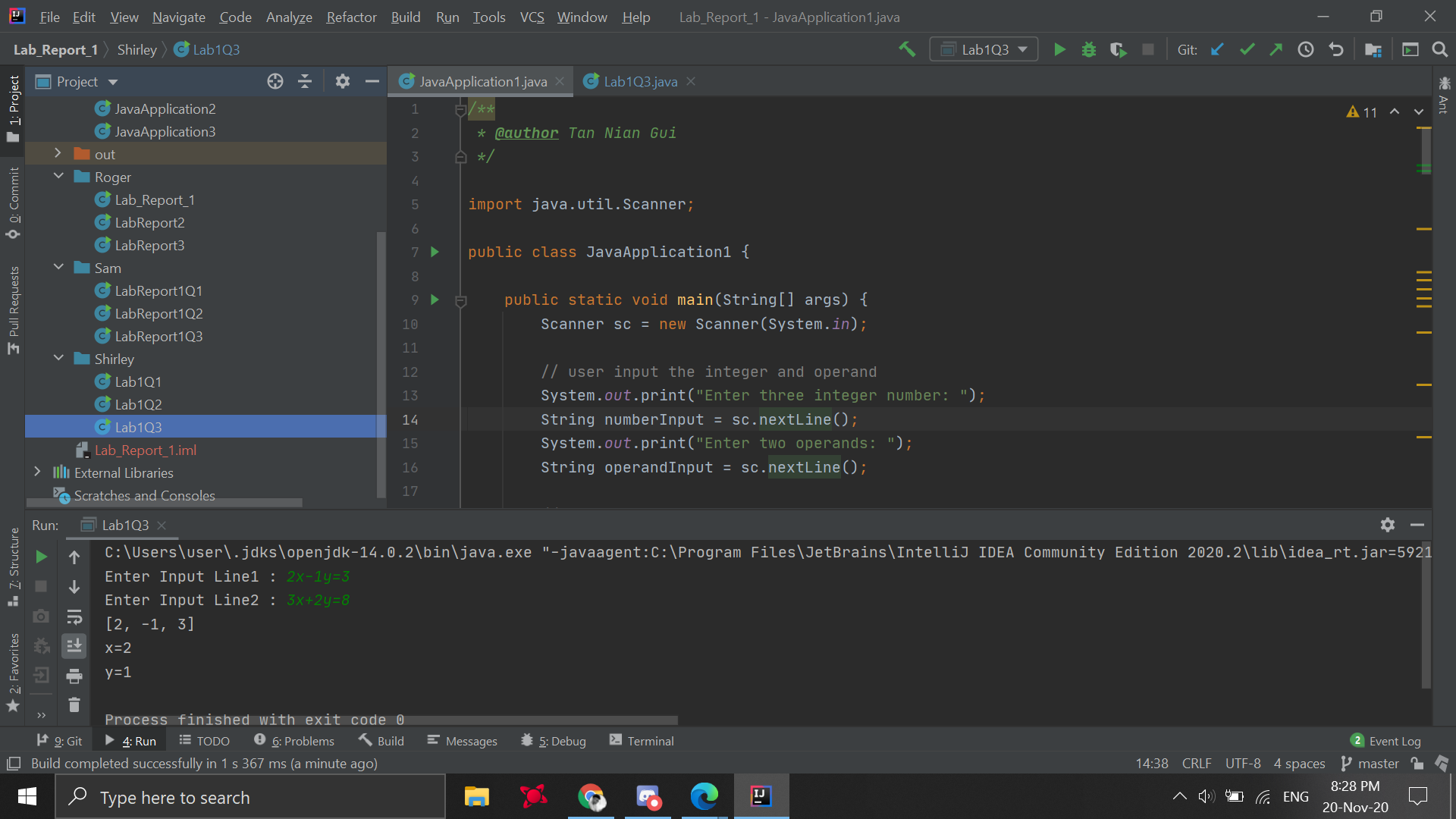
Example 1

Sample Input:



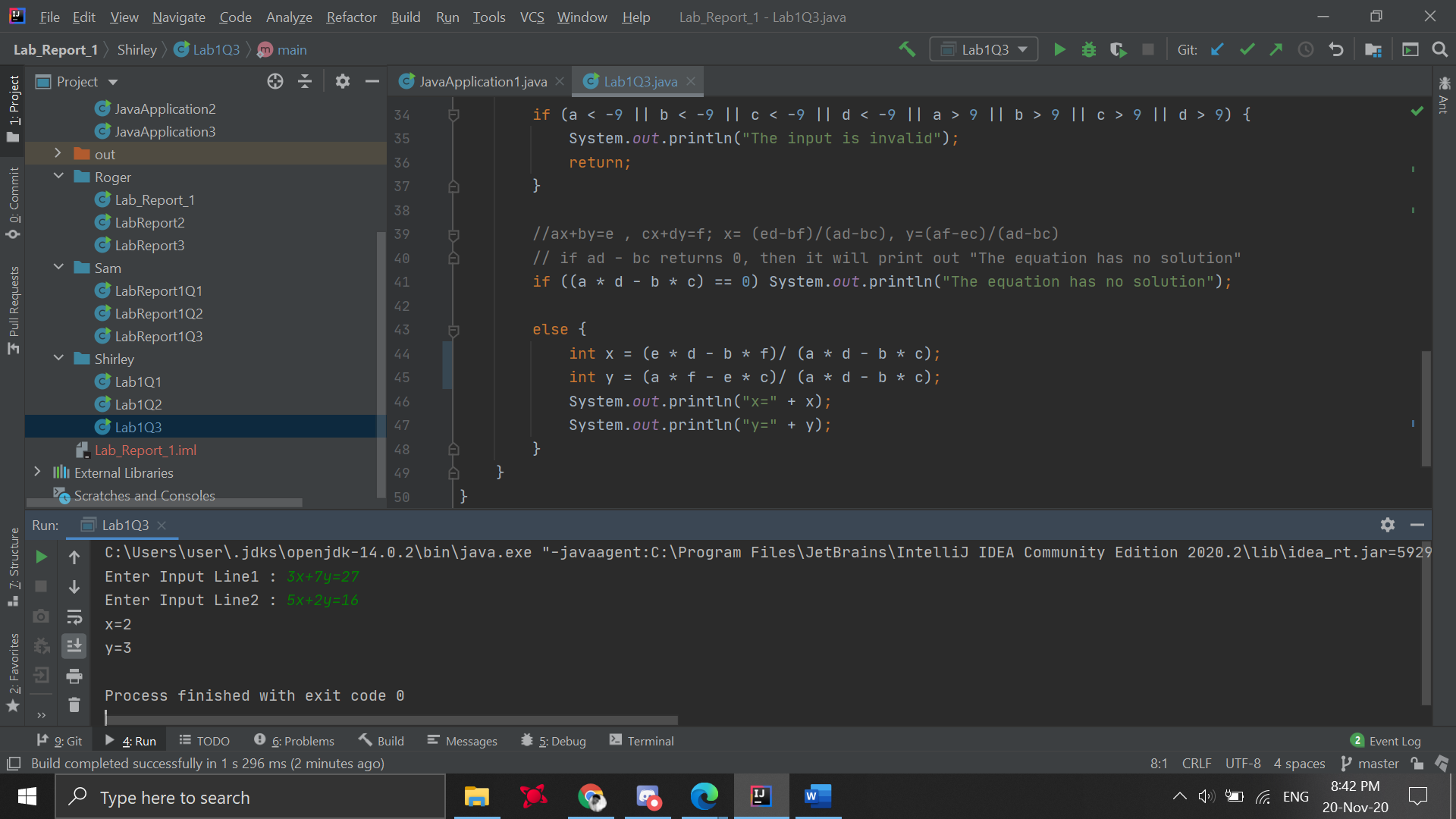
Expected output : x=2, y=1

Actual Output:



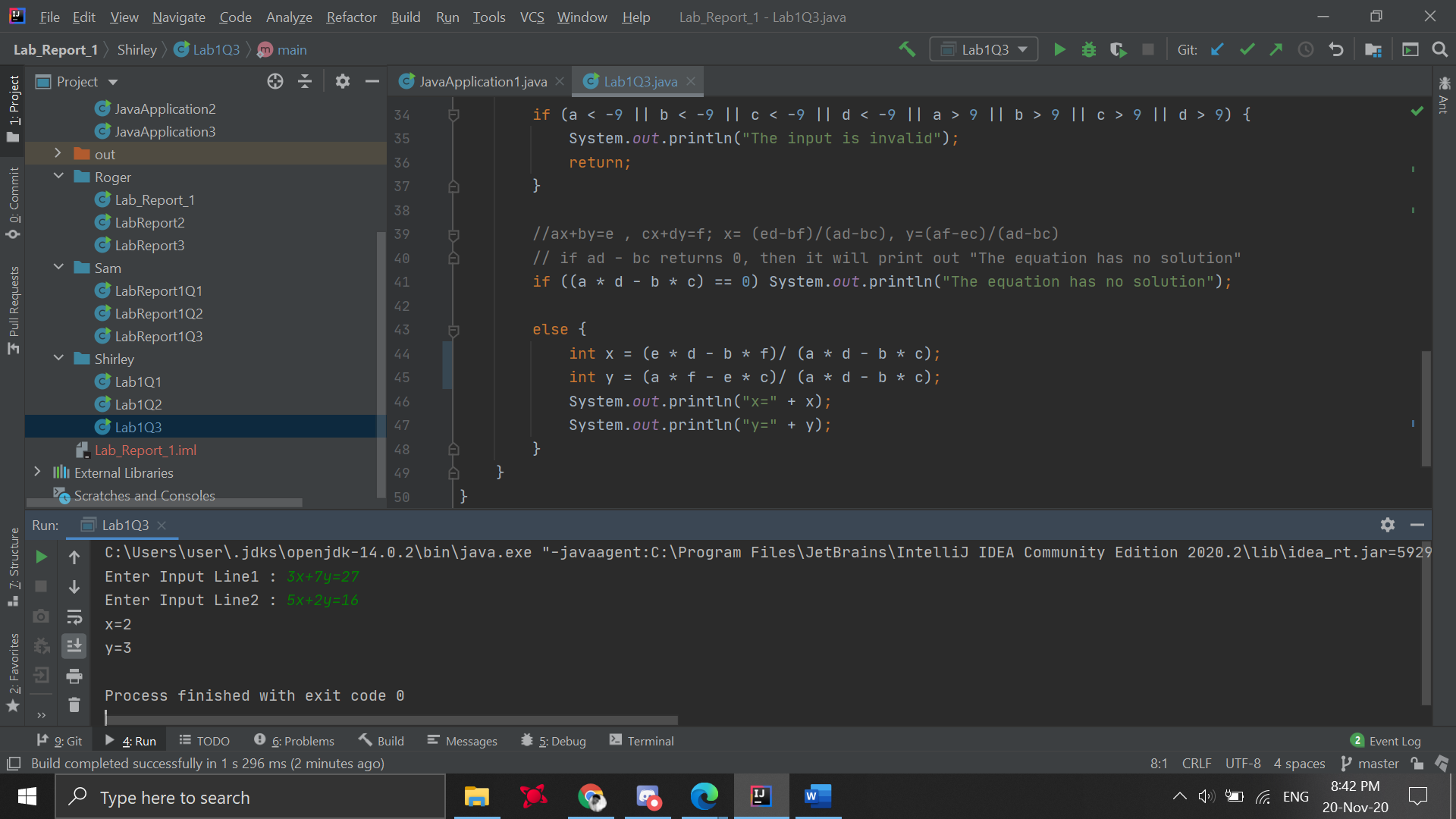
Example 2

Sample Input:



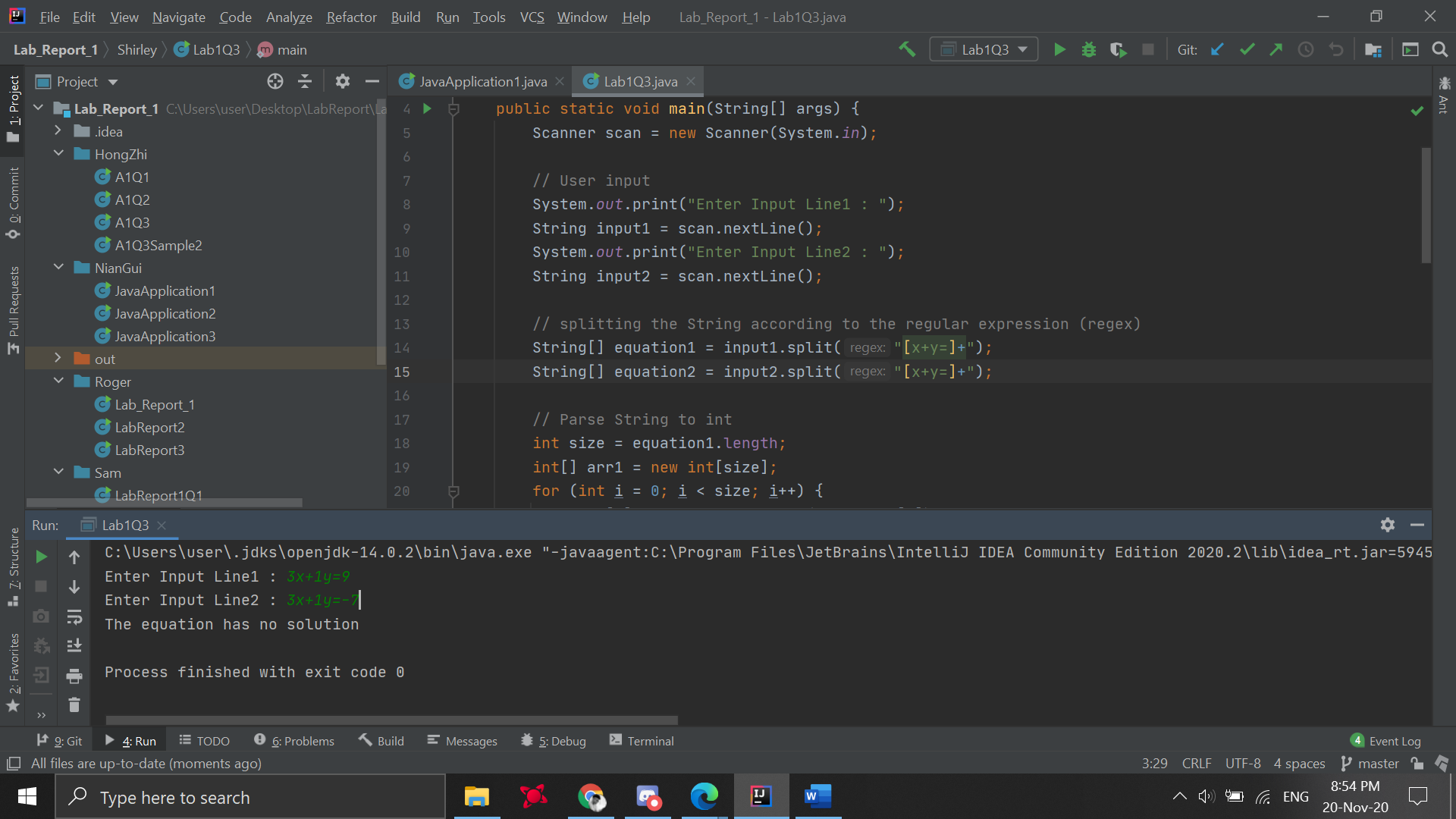
Expected Output: x=2, y=3

Actual Output:



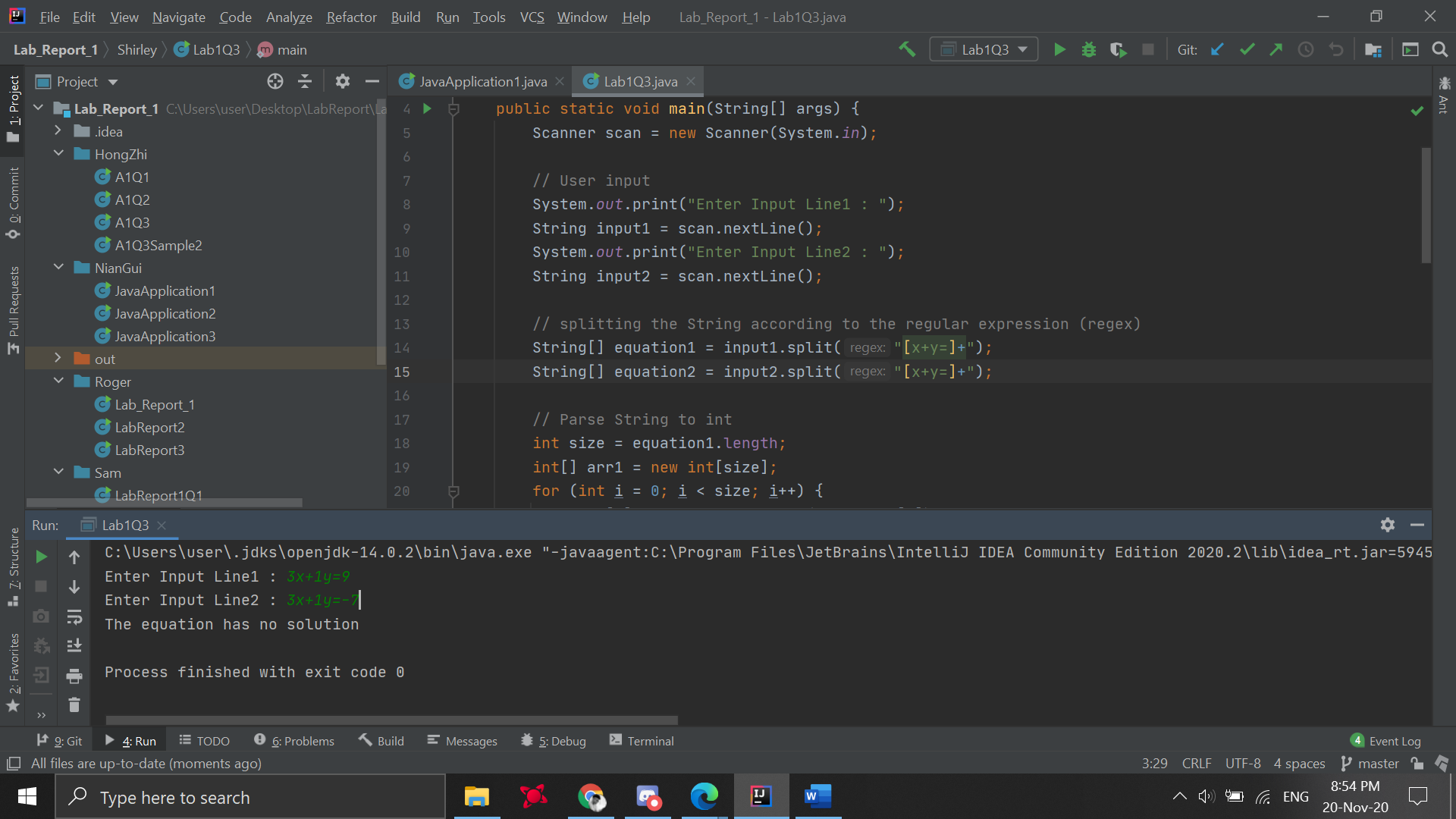
Example 3

Sample Input:



Expected Output: This equation has no solution

Actual Output:



**Source Code**

public class Lab1Q3 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
  
 // User input  
 System.*out*.print("Enter Input Line1 : ");  
 String input1 = scan.nextLine();  
 System.*out*.print("Enter Input Line2 : ");  
 String input2 = scan.nextLine();  
  
 // splitting the String according to the regular expression (regex)  
 String[] equation1 = input1.split("[x+y=]+");  
 String[] equation2 = input2.split("[x+y=]+");  
  
 // Parse String to int  
 int size = equation1.length;  
 int[] arr1 = new int[size];  
 for (int i = 0; i < size; i++) {  
 arr1[i] = Integer.*parseInt*(equation1[i]);  
 }  
  
 int size2 = equation2.length;  
 int[] arr2 = new int[size2];  
 for (int i = 0; i < size2; i++) {  
 arr2[i] = Integer.*parseInt*(equation2[i]);  
 }  
  
 // follow the equation to declare the variables  
 int a = arr1[0], b= arr1[1], e = arr1[2];  
 int c = arr2[0], d = arr2[1], f = arr2[2];  
  
 if (a < -9 || b < -9 || c < -9 || d < -9 || a > 9 || b > 9 || c > 9 || d > 9) {  
 System.*out*.println("The input is invalid");  
 return;  
 }  
  
 //ax+by=e , cx+dy=f; x= (ed-bf)/(ad-bc), y=(af-ec)/(ad-bc)  
 // if ad - bc returns 0, then it will print out "The equation has no solution"  
 if ((a \* d - b \* c) == 0) System.*out*.println("The equation has no solution");  
  
 else {  
 int x = (e \* d - b \* f)/ (a \* d - b \* c);  
 int y = (a \* f - e \* c)/ (a \* d - b \* c);  
 System.*out*.println("x=" + x);  
 System.*out*.println("y=" + y);  
 }  
 }  
}