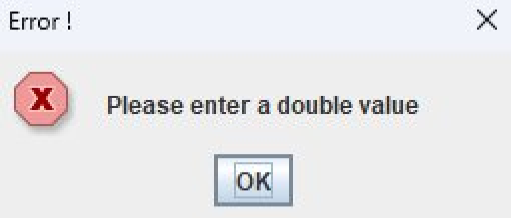
Lab 1. Environment Setup & Java Basics

Họ và tên: Đỗ Thành Trung **MSSV: 202416833**

Note:

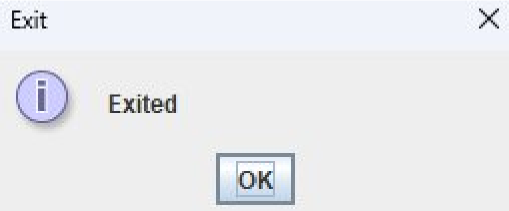
- **All of my programs have cases when the input is non-numeric or incompatible with the requirements:**

  Case 1: When the input is non-numeric or the input is empty



Please enter a double value

  Case 2: When the user exits the program



Exited

Bài 2.2.5. Write a program to calculate sum, difference, product, and quotient of 2 double numbers which are entered by users.

Notes

- To convert from String to double, you can use

```
double num1 = Double.parseDouble(strNum1)
```

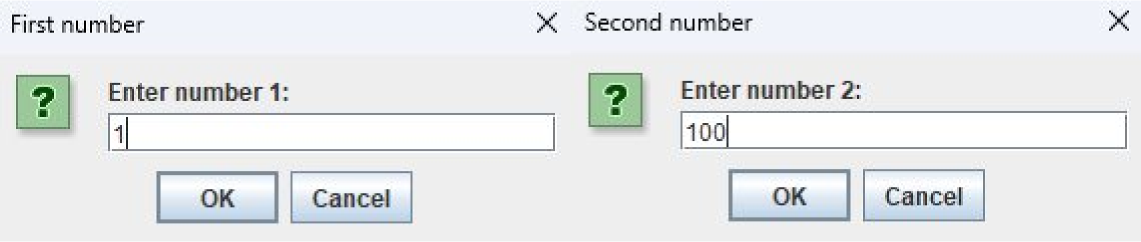
- Check the divisor of the division
- Don't forget to add and commit your work using **'git add .'** and **'git commit -m <message>'** command

1. Code

```
import javax.swing.JOptionPane;

public class Bai225ArithmeticOperations {
    public static void main(String[] args) {
        try {
            String strNum1 = JOptionPane.showInputDialog(null, "Enter number 1:", "First number", JOptionPane.QUESTION_MESSAGE);
            if (strNum1 == null) {
                JOptionPane.showMessageDialog(null, "Exited", "Exit", JOptionPane.INFORMATION_MESSAGE);
                return;
            }
            String strNum2 = JOptionPane.showInputDialog(null, "Enter number 2:", "Second number", JOptionPane.QUESTION_MESSAGE);
            if (strNum2 == null) {
                JOptionPane.showMessageDialog(null, "Exited", "Exit", JOptionPane.INFORMATION_MESSAGE);
                return;
            }

            double firstNum = Double.parseDouble(strNum1);
            double secondNum = Double.parseDouble(strNum2);

            double sum = firstNum + secondNum;
            double difference = Math.abs(firstNum - secondNum);
            double product = firstNum * secondNum;

            String result = "Results:\n"
                    + "Sum = " + sum + "\n"
                    + "Difference = " + difference + "\n"
                    + "Product = " + product + "\n";

            if (secondNum != 0) {
                double quotient = firstNum / secondNum;
                result += "Quotient = " + quotient;
            } else {
                result += "Division by zero is not allowed!";
            }

            JOptionPane.showMessageDialog(null, result, "Result", JOptionPane.INFORMATION_MESSAGE);

        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null, "Please enter a double value", "Error !", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

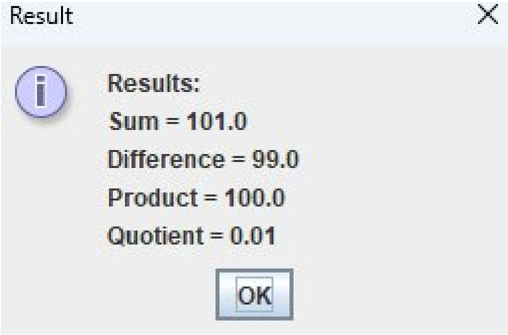
2. Run

Case 1: Normal case



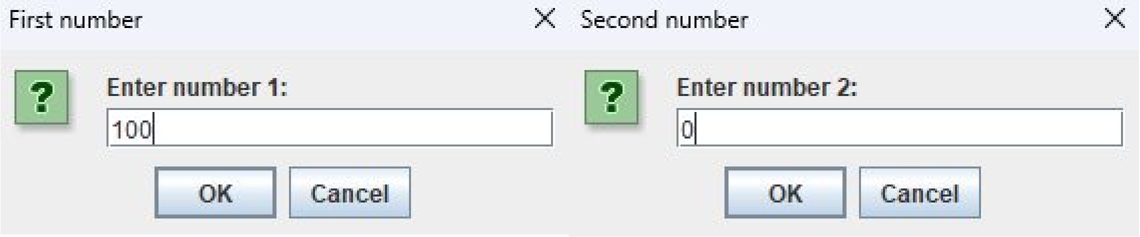
Enter number 1: 1

Enter number 2: 100



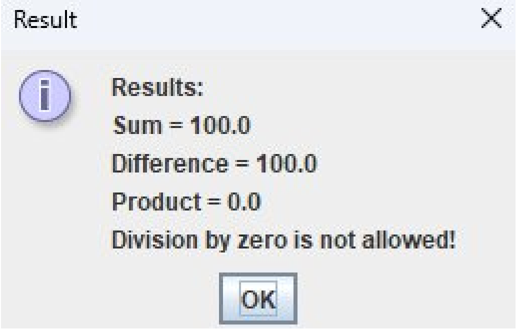
Results:
Sum = 101.0
Difference = 99.0
Product = 100.0
Quotient = 0.01

Case 2: When the second number is zero



Enter number 1: 100

Enter number 2: 0



Results:
Sum = 100.0
Difference = 100.0
Product = 0.0
Division by zero is not allowed!

Bài 2.2.6. Write a program to solve

For simplicity, we only consider the real roots of the equations in this task.

3. The first-degree equation (linear equation) with one variable

Note: A first-degree equation with one variable can have a form such as $ax + b = 0$ $(a \neq$

$0)$. You should handle the case where the user input value 0 for a.

Don't forget to add and commit your work using '**git add .**' and '**git commit -m <message>**' command

4. The system of first-degree equations (linear system) with two variables

Note: A system of first-degree equations with two variables x_1 and x_2 can be written as follows.

$$\{a_{11}x_1 + a_{12}x_2 = b_1\ a_{21}x_1 + a_{22}x_2 = b_2$$

You should handle the case where the values of the coefficients produce infinitely many solutions and the case where they produce no solution.

Hint:

Use the following determinants:

$$D = |a_{11}\ a_{12}\ a_{21}\ a_{22}\ | = a_{11}a_{22} - a_{21}a_{12}D_1 = |b_1\ a_{12}\ b_2\ a_{22}\ | = b_1a_{22} - b_2a_{12}D_2 = |a_{11}\ b_1\ a_{21}\ b_2\ | = a_{11}b_2 - a_{21}b_1$$

5. The second-degree equation with one variable

Note: A second-degree equation with one variable (i.e., quadratic equation) can have a form such as $ax^2 +$
$bx + c = 0$, where x is the variable, and a, b, and c are coefficients $(a \neq 0)$.

You should handle the case where the values of the coefficients produce a double root & the case where they produce no root. You should also handle the case where the user input value 0 for a.

Hint:

Use the discriminant $\Delta = b^2 - 4ac$

After completing the code in section 2, you should push all of your changes, including assignment 2.2.1 to 2.2.6 to the **master** branch of the valid repository you have created.

Hint: You should use "git push origin <name of the branch that you want to push to the remote repository>", in this case is **"master"** , to push all of your works to the **master** branch.

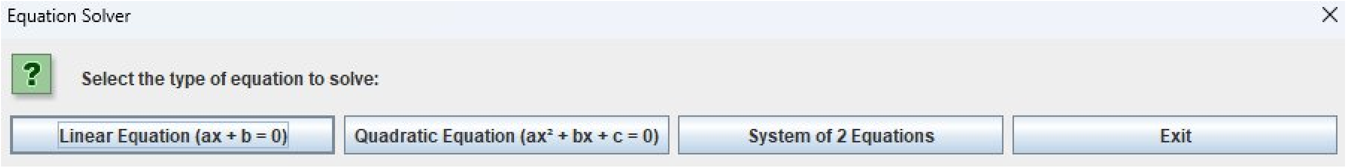
1. Code

```java
import javax.swing.JOptionPane;

public class Bai226EquationSolver {
    public static void main(String[] args) {
        while (true) {
            String[] options = {"Linear Equation (ax + b = 0)", "Quadratic Equation
(ax² + bx + c = 0)", "System of 2 Equations", "Exit"};
            int choice = JOptionPane.showOptionDialog(null,
                    "Select the type of equation to solve:",
                    "Equation Solver",
                    JOptionPane.DEFAULT_OPTION,
                    JOptionPane.QUESTION_MESSAGE,
                    null,
                    options,
                    options[0]);

            if (choice == 3 || choice == JOptionPane.CLOSED_OPTION) {
                JOptionPane.showMessageDialog(null, "Exited!", "Exit",
JOptionPane.INFORMATION_MESSAGE);
                break;
            }

            switch (choice) {
                case 0:
                    solveLinearEquation();
                    break;
                case 1:
                    solveQuadraticEquation();
                    break;
                case 2:
                    solveSystemOfEquations();
                    break;
            }
        }
    }

    public static void solveLinearEquation() {
        try {
            String Coef_a = JOptionPane.showInputDialog(null, "Enter the
coefficient for x:", "Enter a", JOptionPane.QUESTION_MESSAGE);
            if (Coef_a == null) return;
            String Free_coef = JOptionPane.showInputDialog(null, "Enter the free
coefficient:", "Enter b", JOptionPane.QUESTION_MESSAGE);
            if (Free_coef == null) return;

            double a_value = Double.parseDouble(Coef_a);
            double b_value = Double.parseDouble(Free_coef);
            String result = "";

            if (a_value != 0) {
                double x_value = -b_value / a_value;
                result = "The solution for the equation is x = " + x_value;
            } else {
                if (b_value == 0)
```

```
                    result = "Infinitely many solutions";
                else
                    result = "No solution found";
            }

            JOptionPane.showMessageDialog(null, result, "Result",
JOptionPane.INFORMATION_MESSAGE);
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null, "Please enter a double value",
"Error!", JOptionPane.ERROR_MESSAGE);
        }
    }

    public static void solveQuadraticEquation() {
        try {
            String Coef_a = JOptionPane.showInputDialog(null, "Enter the
coefficient for x²:", "Enter a", JOptionPane.QUESTION_MESSAGE);
            if (Coef_a == null) return;
            String Coef_b = JOptionPane.showInputDialog(null, "Enter the
coefficient for x:", "Enter b", JOptionPane.QUESTION_MESSAGE);
            if (Coef_b == null) return;
            String Free_coef = JOptionPane.showInputDialog(null, "Enter the free
coefficient:", "Enter c", JOptionPane.QUESTION_MESSAGE);
            if (Free_coef == null) return;

            double a_value = Double.parseDouble(Coef_a);
            double b_value = Double.parseDouble(Coef_b);
            double c_value = Double.parseDouble(Free_coef);

            String result = "Result:\n";

            if (a_value == 0) {
                if (b_value == 0) {
                    if (c_value == 0) result += "Infinitely many solutions";
                    else result += "No solution found";
                } else {
                    double x = -c_value / b_value;
                    result += "The solution for the equation is x = " + x;
                }
            } else {
                double delta = b_value * b_value - 4 * a_value * c_value;

                if (delta > 0) {
                    double x1 = (-b_value + Math.sqrt(delta)) / (2 * a_value);
                    double x2 = (-b_value - Math.sqrt(delta)) / (2 * a_value);
                    result += "The solutions for the equation are:\n";
                    result += "x1 = " + x1 + "\n";
                    result += "x2 = " + x2;
                } else if (delta == 0) {
                    double x = -b_value / (2 * a_value);
                    result += "The equation has one double root: x = " + x;
                } else {
                    result += "No real solution found";
                }
            }

            JOptionPane.showMessageDialog(null, result, "Result",
JOptionPane.INFORMATION_MESSAGE);
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null, "Please enter a double value",
"Error!", JOptionPane.ERROR_MESSAGE);
        }
    }

    public static void solveSystemOfEquations() {
        try {
            String s_a11 = JOptionPane.showInputDialog(null, "Enter a11:");
            if (s_a11 == null) return;
            String s_a12 = JOptionPane.showInputDialog(null, "Enter a12:");
            if (s_a12 == null) return;
            String s_b1 = JOptionPane.showInputDialog(null, "Enter b1:");
            if (s_b1 == null) return;
            String s_a21 = JOptionPane.showInputDialog(null, "Enter a21:");
            if (s_a21 == null) return;
            String s_a22 = JOptionPane.showInputDialog(null, "Enter a22:");
            if (s_a22 == null) return;
            String s_b2 = JOptionPane.showInputDialog(null, "Enter b2:");
            if (s_b2 == null) return;

            double a11 = Double.parseDouble(s_a11);
            double a12 = Double.parseDouble(s_a12);
            double b1 = Double.parseDouble(s_b1);
            double a21 = Double.parseDouble(s_a21);
            double a22 = Double.parseDouble(s_a22);
            double b2 = Double.parseDouble(s_b2);

            double D = a11 * a22 - a12 * a21;
            double Dx = b1 * a22 - a12 * b2;
            double Dy = a11 * b2 - b1 * a21;

            String result;

            if (D != 0) {
                double x = Dx / D;
                double y = Dy / D;
                result = "The solutions for the set of equations are:\n" +
                        "x = " + x + "\n" +
                        "y = " + y;
            } else {
                if (Dx == 0 && Dy == 0)
                    result = "Infinitely many solutions";
                else
                    result = "No solution found";
            }

            JOptionPane.showMessageDialog(null, result, "Result",
JOptionPane.INFORMATION_MESSAGE);
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null, "Please enter a double value",
"Error!", JOptionPane.ERROR_MESSAGE);
        }
  }
```

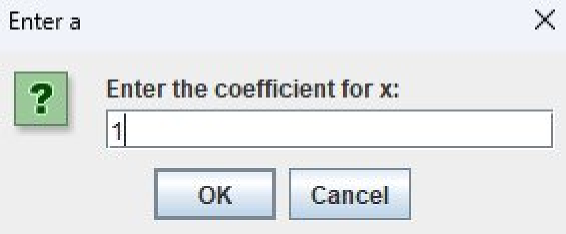
2. Run

This is the menu when we start the program


Equation Solver
Linear Equation (ax + b = 0)
Quadratic Equation ($ax^2 + bx + c = 0$)
System of 2 Equations
Exit

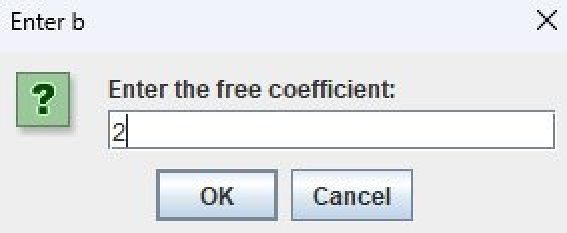
Select the type of equation to solve:

Case 1: Linear equation normal case


Enter a

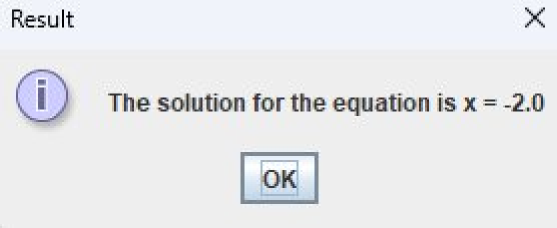
Enter the coefficient for x:

1


Enter b

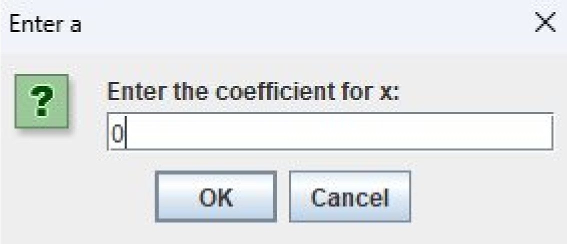
Enter the free coefficient:

2


Result

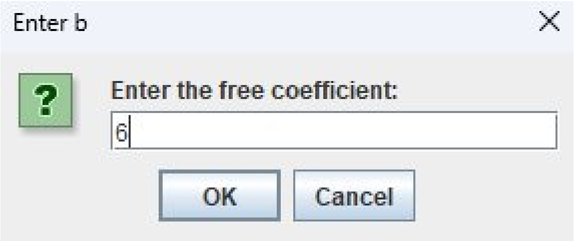
The solution for the equation is x = -2.0

Case 2: Linear equation when a = 0


Enter a

Enter the coefficient for x:

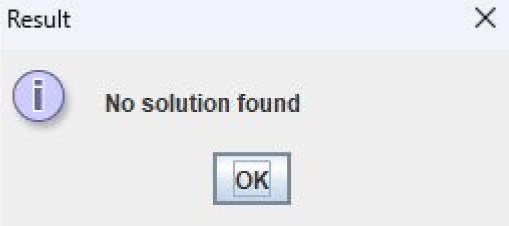
0


Enter b

Enter the free coefficient:

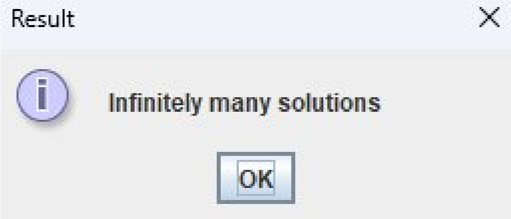
6

When b != 0 the result is


Result

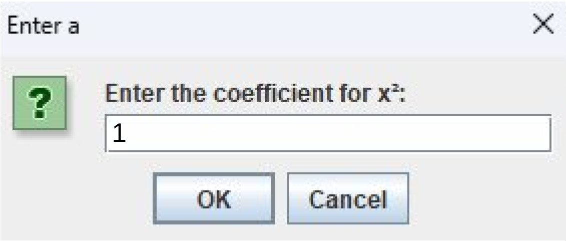
No solution found

Else the result will be


Result

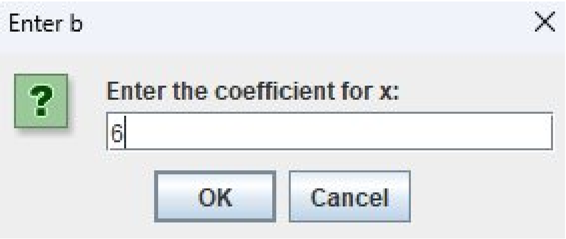
Infinitely many solutions

Case 3: Quadratic equation normal case



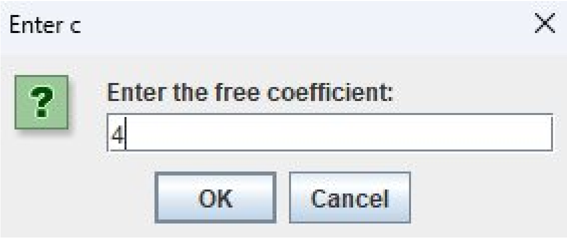
Enter the coefficient for x^2:

1



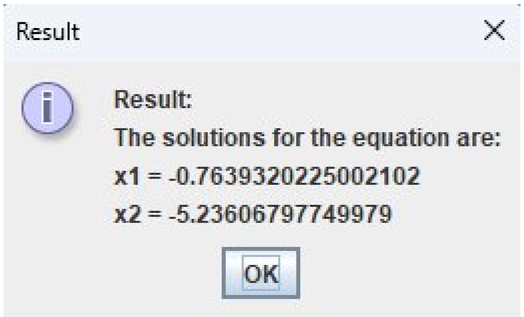
Enter the coefficient for x:

6



Enter the free coefficient:

4

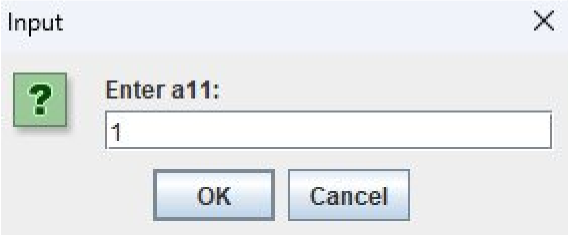


```
Result:
The solutions for the equation are:
x1 = -0.7639320225002102
x2 = -5.23606797749979
```

Case 4: When a = 0, the equation will turn linear which is case 2 in this exercise

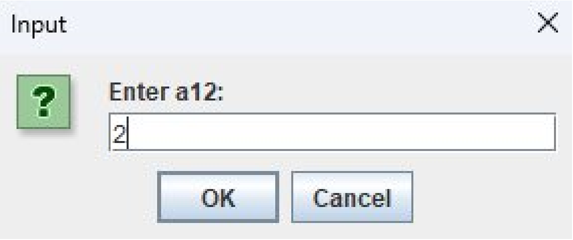
Case 5: A set of two linear equations

In this case we have the example of this set of equations: $\begin{cases} x + 2y = 10 \\ 2x + 8y = 8 \end{cases}$



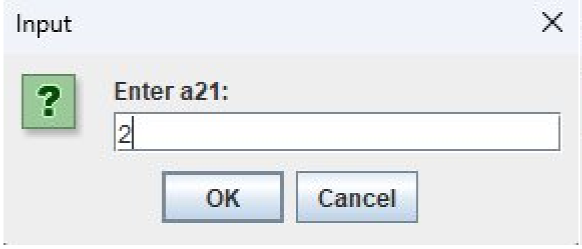
Enter a11:

1



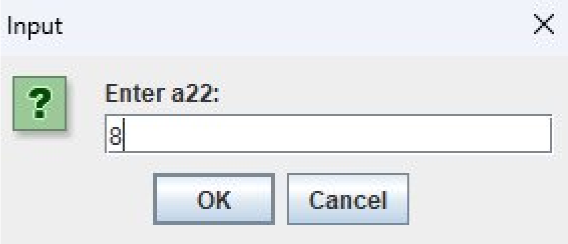
Enter a12:

2



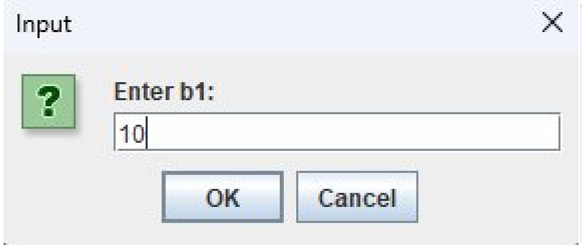
Enter a21:

2



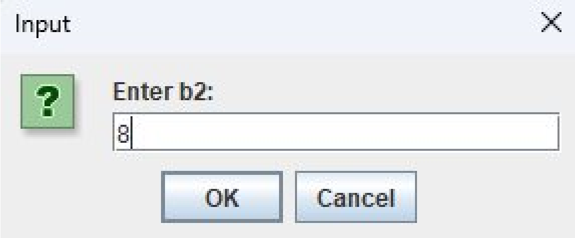
Enter a22:

8



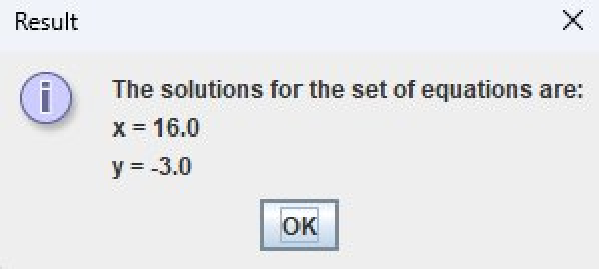
Enter b1:

10



Enter b2:

8

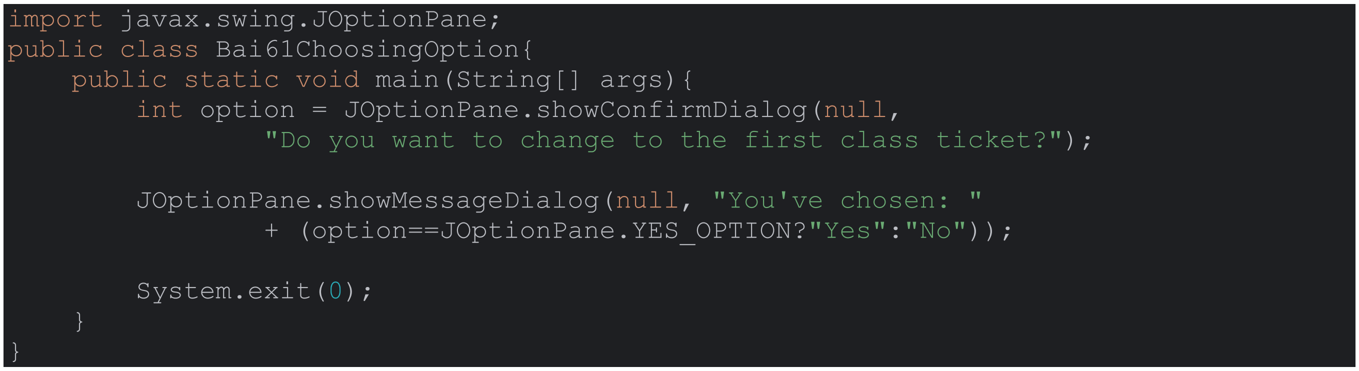


```
The solutions for the set of equations are:
x = 16.0
y = -3.0
```

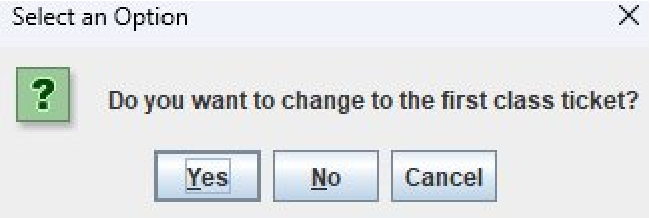
The program also works well in the cases of infinitely many solutions or no solution found

Bài 6.1. Write, compile and run the ChoosingOption program

1. Code and Run



```java
import javax.swing.JOptionPane;
public class Bai61ChoosingOption{
    public static void main(String[] args){
        int option = JOptionPane.showConfirmDialog(null,
                "Do you want to change to the first class ticket?");

        JOptionPane.showMessageDialog(null, "You've chosen: "
                + (option==JOptionPane.YES_OPTION?"Yes":"No"));

        System.exit(0);
    }
}
```

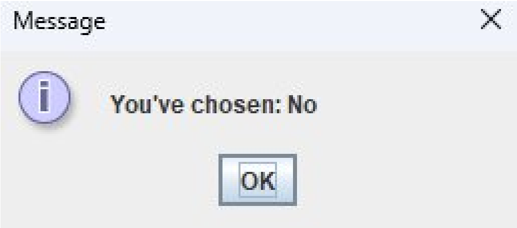


Select an Option

Do you want to change to the first class ticket?

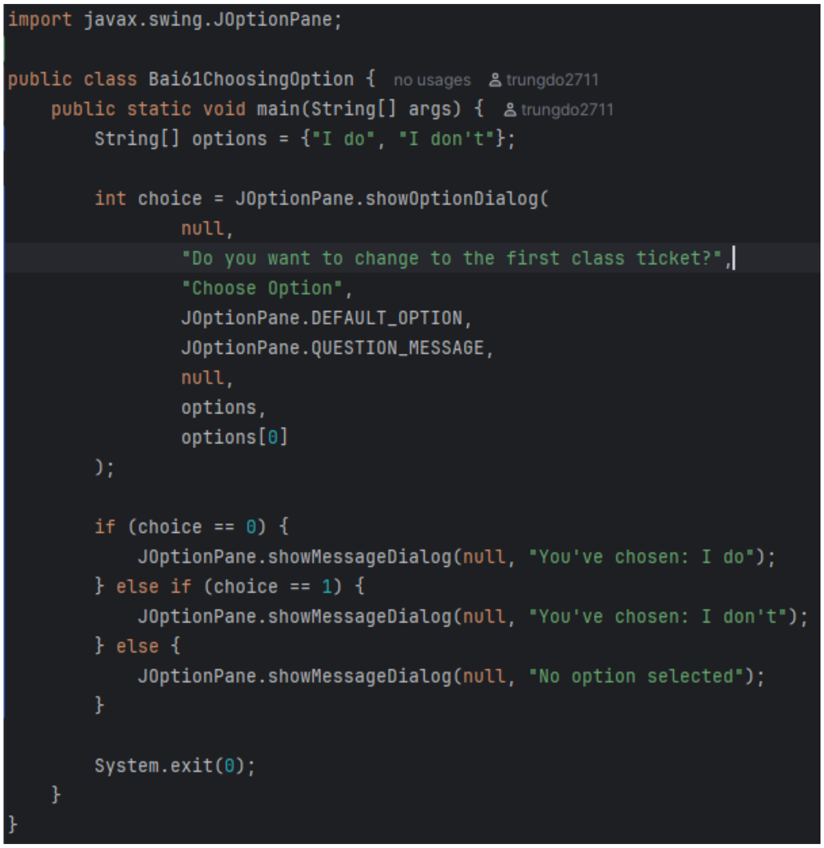
2. Question

- What happens if users choose "Cancel"?
    - If users choose cancel, the message will be you've chosen No, which is incorrect



You've chosen: No

- To customize the options to e.g: "I do" or "I don't", instead of using showConfirmDialog, we use showOptionDialog. Afterward, we will be able to set up the program's output to "I do" or "I don't" using several lines of code.
-

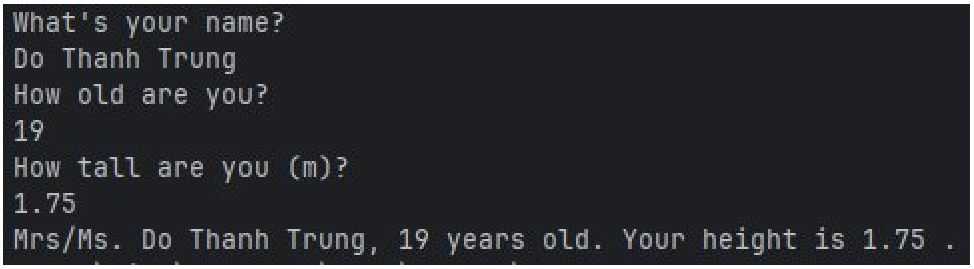


```
import javax.swing.JOptionPane;

public class Bai61ChoosingOption {  no usages  trungdo2711
    public static void main(String[] args) {  trungdo2711
        String[] options = {"I do", "I don't"};

        int choice = JOptionPane.showOptionDialog(
                null,
                "Do you want to change to the first class ticket?",
                "Choose Option",
                JOptionPane.DEFAULT_OPTION,
                JOptionPane.QUESTION_MESSAGE,
                null,
                options,
                options[0]
        );

        if (choice == 0) {
            JOptionPane.showMessageDialog(null, "You've chosen: I do");
        } else if (choice == 1) {
            JOptionPane.showMessageDialog(null, "You've chosen: I don't");
        } else {
            JOptionPane.showMessageDialog(null, "No option selected");
        }

        System.exit(0);
    }
}
```

Bài 6.2. Write a program for input/output from keyboard

Code – Compile – Run

```
import java.util.Scanner;
public class Bai62InputFromKeyboard{
    public static void main(String args[]){
        Scanner keyboard = new Scanner(System.in);

        System.out.println("What's your name?");
        String strName = keyboard.nextLine();
        System.out.println("How old are you?");
        int iAge = keyboard.nextInt();
        System.out.println("How tall are you (m)?");
        double dHeight = keyboard.nextDouble();

        System.out.println("Mrs/Ms. " + strName + ", " + iAge + " years old. "
                + "Your height is " + dHeight + " .");
    }
}
```



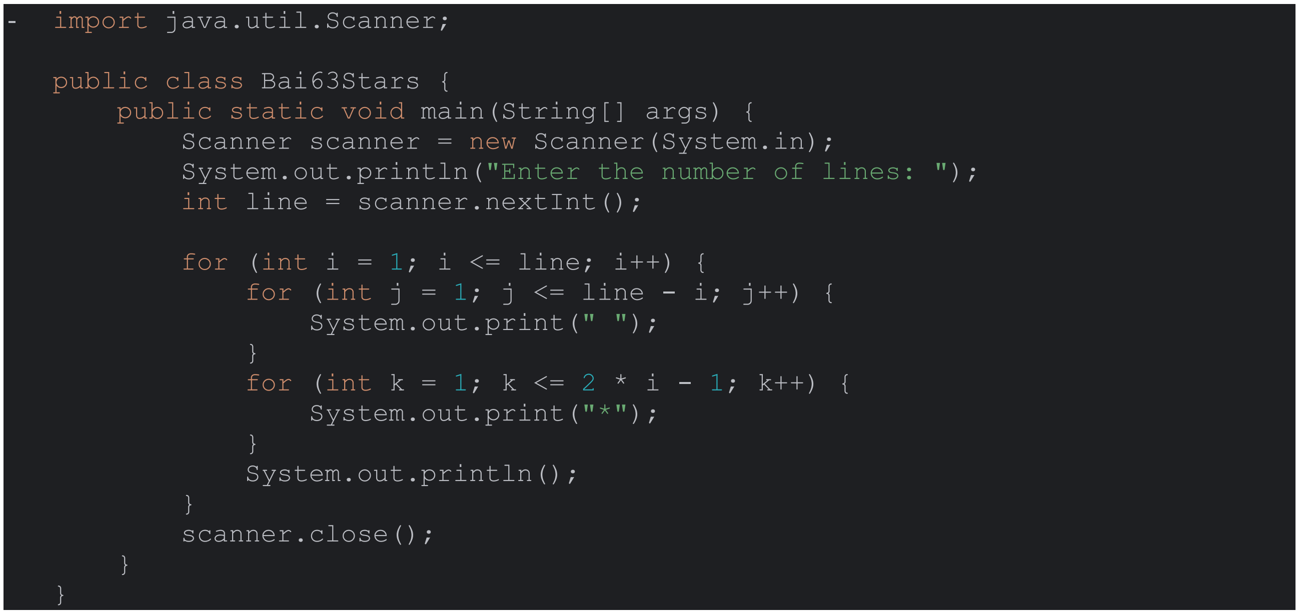
```
What's your name?
Do Thanh Trung
How old are you?
19
How tall are you (m)?
1.75
Mrs/Ms. Do Thanh Trung, 19 years old. Your height is 1.75 .
```

Bài 6.3. Write a program to display a triangle with a height of n stars (*), n is entered by users.

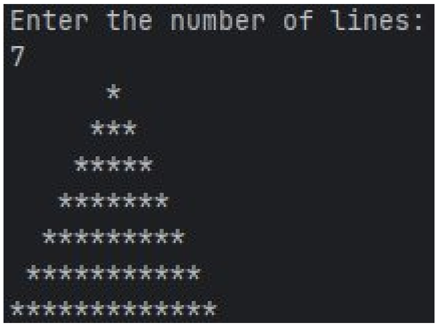
E.g. n = 5:

```
    *
   ***
  *****
 *******
*********
```

- **Code & Run**



```
-   import java.util.Scanner;

    public class Bai63Stars {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.println("Enter the number of lines: ");
            int line = scanner.nextInt();

            for (int i = 1; i <= line; i++) {
                for (int j = 1; j <= line - i; j++) {
                    System.out.print(" ");
                }
                for (int k = 1; k <= 2 * i - 1; k++) {
                    System.out.print("*");
                }
                System.out.println();
            }
            scanner.close();
        }
    }
```



```
Enter the number of lines:
7
      *
     ***
    *****
   *******
  *********
 ***********
*************
```

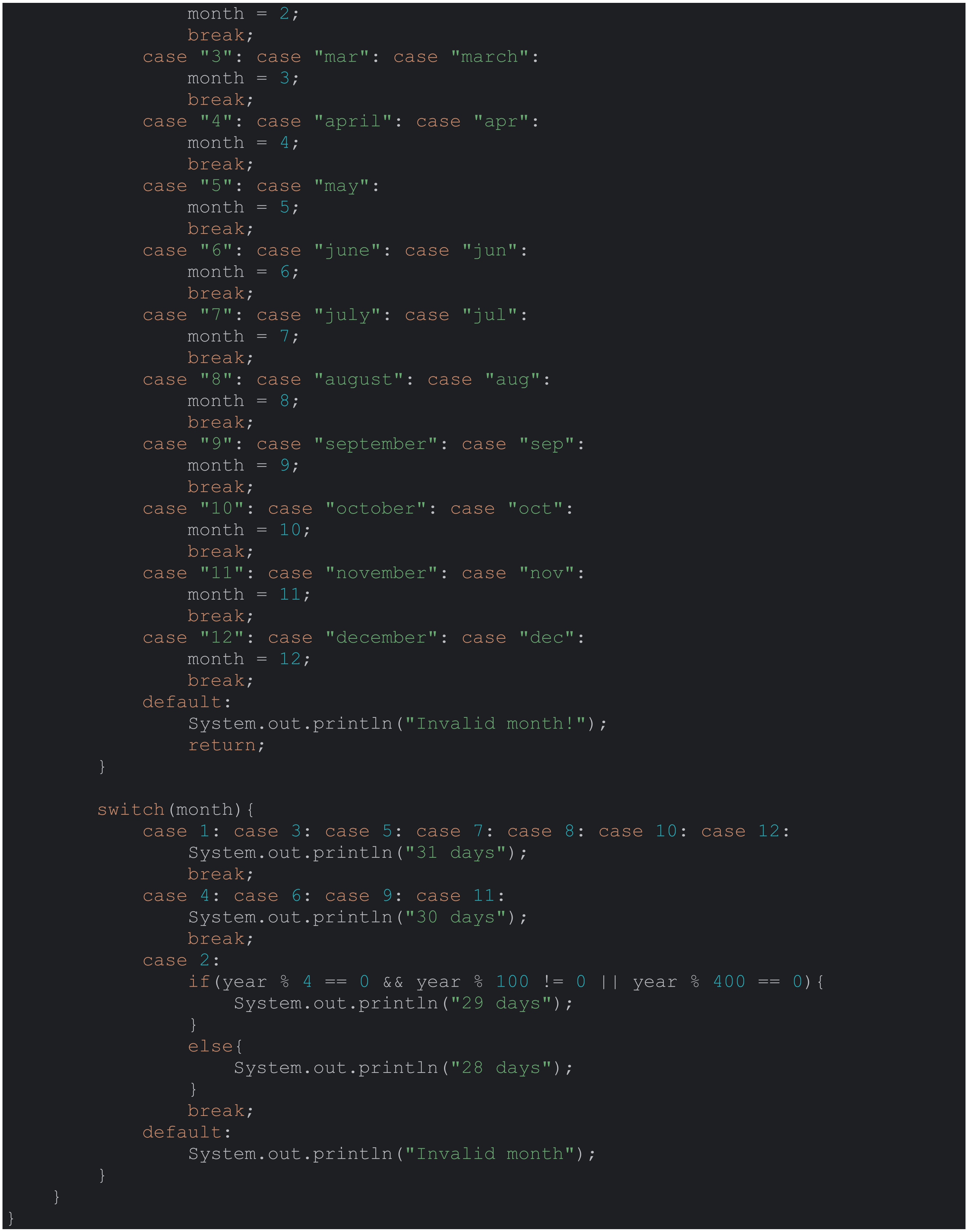
Bài 6.4. Write a program to display the number of days of a month, which is entered by users (both month and year). If it is an invalid month/year, ask the user to enter again.

- The user can either enter a month in its full name, abbreviation, in 3 letters, or in number. To illustrate, the valid inputs of *January* are January, Jan., Jan, and 1.
- The user must enter a year in a non-negative number and enter all the digits. For instance, the valid input of year *1999* is only 1999, but not 99, "one thousand nine hundred ninety-nine", or anything else.
- A year is either a common year of 365 days or a leap year of 366 days. Every year that is divisible by 4 is a leap year, except for years that are divisible by 100, but not by 400. For instance, the year 1800 is not a leap year, yet the year 2000 is a leap year. In a year, there are twelve months, which are listed in order as follows.

Month	January	February	March	April	May	June	July	August	September	October	November	December
Abbreviation	Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.
In 3 letters	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
In Number	1	2	3	4	5	6	7	8	9	10	11	12
Days of Month in Common Year	31	28	31	30	31	30	31	31	30	31	30	31
Days of Month in Leap Year	31	**29**	31	30	31	30	31	31	30	31	30	31

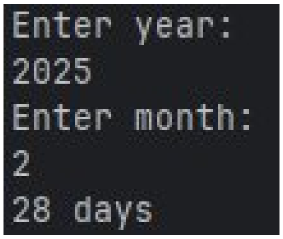
- **Code**

```java
import java.util.Scanner;
public class Bai64MonthProgram {
    public static void main(String[] args){
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter year: ");
        int year = scanner.nextInt();
        scanner.nextLine();
        System.out.println("Enter month: ");
        String input = scanner.nextLine().trim();
        input = input.replace(".", "").toLowerCase();

        int month;

        switch(input){
            case "1": case "jan": case "january":
                month = 1;
                break;
            case "2": case "feb": case "february":
```



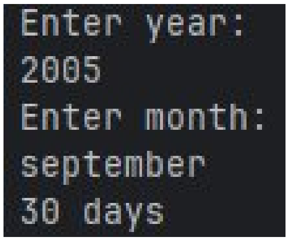
```
                month = 2;
                break;
            case "3": case "mar": case "march":
                month = 3;
                break;
            case "4": case "april": case "apr":
                month = 4;
                break;
            case "5": case "may":
                month = 5;
                break;
            case "6": case "june": case "jun":
                month = 6;
                break;
            case "7": case "july": case "jul":
                month = 7;
                break;
            case "8": case "august": case "aug":
                month = 8;
                break;
            case "9": case "september": case "sep":
                month = 9;
                break;
            case "10": case "october": case "oct":
                month = 10;
                break;
            case "11": case "november": case "nov":
                month = 11;
                break;
            case "12": case "december": case "dec":
                month = 12;
                break;
            default:
                System.out.println("Invalid month!");
                return;
        }

        switch(month){
            case 1: case 3: case 5: case 7: case 8: case 10: case 12:
                System.out.println("31 days");
                break;
            case 4: case 6: case 9: case 11:
                System.out.println("30 days");
                break;
            case 2:
                if(year % 4 == 0 && year % 100 != 0 || year % 400 == 0){
                    System.out.println("29 days");
                }
                else{
                    System.out.println("28 days");
                }
                break;
            default:
                System.out.println("Invalid month");
        }
    }
}
```

- Test 1: Normal case



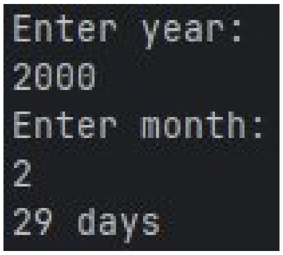
```
Enter year:
2025
Enter month:
2
28 days
```

- Test 2: When the input is a string



```
Enter year:
2005
Enter month:
september
30 days
```

- Test 3: Leap year



```
Enter year:
2000
Enter month:
2
29 days
```

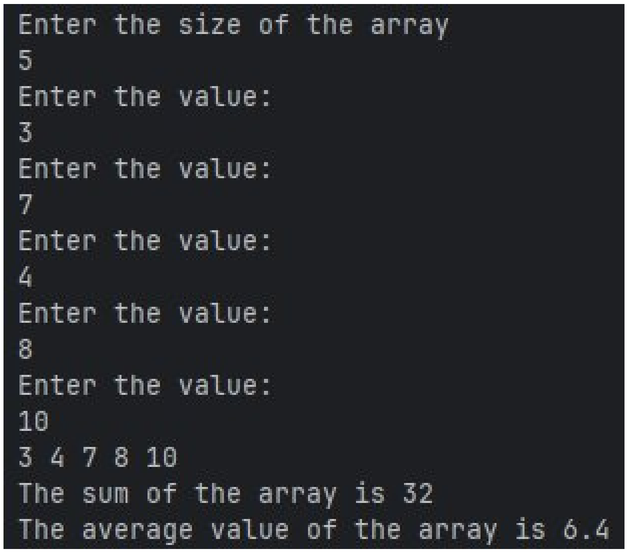
Bài 6.5. Write a Java program to sort a numeric array, and calculate the sum and average value of array elements.

- **Code**

```
-   import java.util.Scanner;

    public class Bai65Array {

        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.println("Enter the size of the array");
            int size = scanner.nextInt();
            int[] array = new int[size];
            for(int i = 0; i < size; i++){
                System.out.println("Enter the value: ");
                int value = scanner.nextInt();
                array[i] = value;
            }
            sortArray(array, 0, size - 1);
            for(int i = 0; i < size; i++){
                System.out.print(array[i] + " ");
            }
            System.out.println();
            System.out.println("The sum of the array is " + calculateSum(array));
            System.out.println("The average value of the array is " +
    calculateAverage(array));
            scanner.close();
        }
        public static int calculateSum(int array[]){
            int sum = 0;
            for(int i = 0; i < array.length; i++){
                sum += array[i];
            }
            return sum;
        }
        public static double calculateAverage(int array[]){
            int sum = 0;
            for(int i = 0; i < array.length; i++){
                sum += array[i];
            }
            return (double) sum / array.length;
        }
        public static void sortArray(int[] array, int left, int right){
            if(left < right){
                int mid = (left + right) / 2;
                sortArray(array, left, mid);
                sortArray(array, mid + 1, right);
                merge(array,left,mid,right);
            }
        }
        public static void merge(int[] array, int left, int mid, int right){
```

```
        int size1 = mid - left + 1;
        int size2 = right - mid;

        int[] L = new int[size1];
        int[] R = new int[size2];

        for(int i = 0; i < size1; i++){
            L[i] = array[left + i];
        }
        for(int j = 0; j < size2; j++){
            R[j] = array[mid + 1 + j];
        }
        int i = 0, j = 0;
        int k = left;

        while(i < size1 && j < size2){
            if(L[i] <= R[j]){
                array[k] = L[i];
                i++;
            }else{
                array[k] = R[j];
                j++;
            }
            k++;
        }
        while(i < size1){
            array[k] = L[i];
            i++;
            k++;
        }
        while(j < size2){
            array[k] = R[j];
            j++;
            k++;
        }
    }
}
```

- **Run**

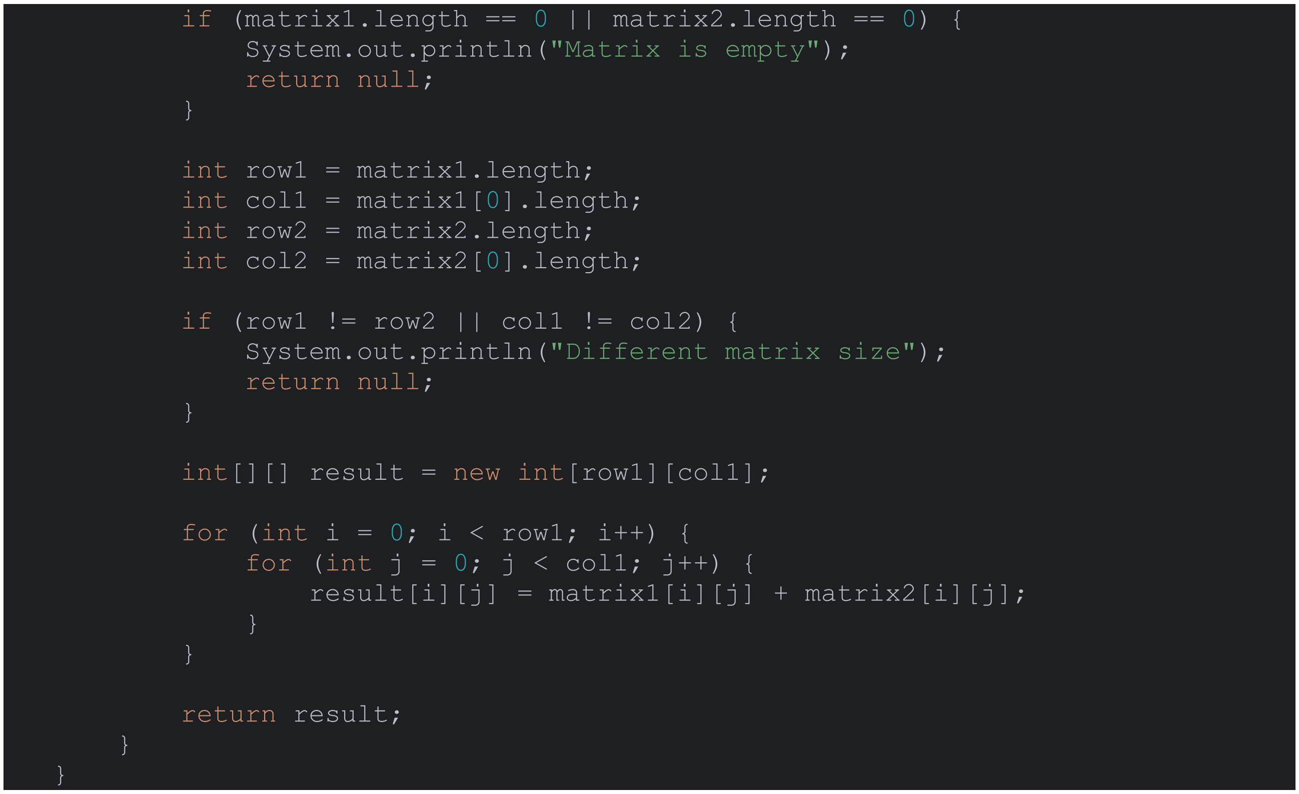


```
Enter the size of the array
5
Enter the value:
3
Enter the value:
7
Enter the value:
4
Enter the value:
8
Enter the value:
10
3 4 7 8 10
The sum of the array is 32
The average value of the array is 6.4
```

Bài 6.6. Write a Java program to add two matrices of the same size.

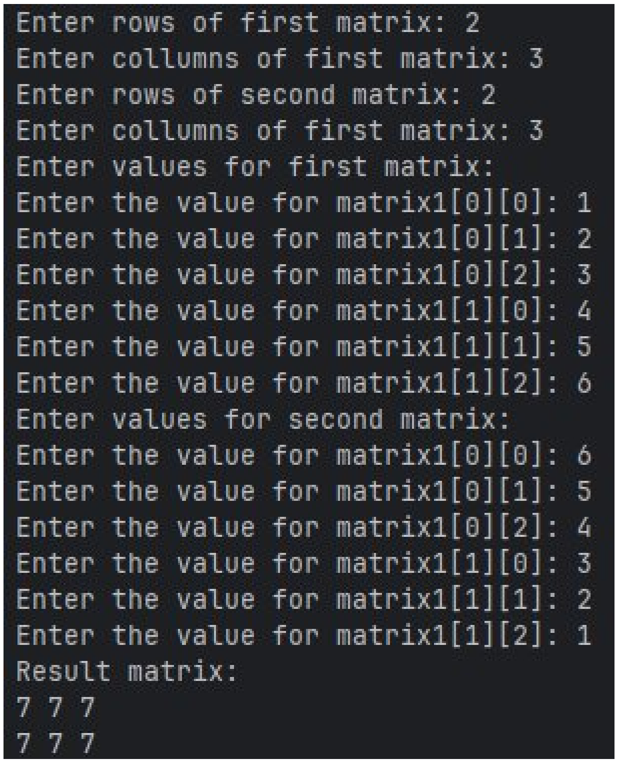
- **Code**

```
-   import java.util.Scanner;

    public class Bai66MatrixAdding {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);

            System.out.print("Enter rows of first matrix: ");
            int row1 = scanner.nextInt();
            System.out.print("Enter collumns of first matrix: ");
            int col1 = scanner.nextInt();
            int[][] matrix1 = new int[row1][col1];

            System.out.print("Enter rows of second matrix: ");
            int row2 = scanner.nextInt();
            System.out.print("Enter collumns of first matrix: ");
            int col2 = scanner.nextInt();
            int[][] matrix2 = new int[row2][col2];

            System.out.println("Enter values for first matrix:");
            for (int i = 0; i < row1; i++) {
                for (int j = 0; j < col1; j++) {
                    System.out.print("Enter the value for matrix1[" + i + "][" +
    j + "]: ");
                    matrix1[i][j] = scanner.nextInt();
                }
            }

            System.out.println("Enter values for second matrix:");
            for (int i = 0; i < row2; i++) {
                for (int j = 0; j < col2; j++) {
                    System.out.print("Enter the value for matrix1[" + i + "][" +
    j + "]: ");
                    matrix2[i][j] = scanner.nextInt();
                }
            }

            int[][] result = addingMatrix(matrix1, matrix2);

            if (result != null) {
                System.out.println("Result matrix:");
                for (int i = 0; i < result.length; i++) {
                    for (int j = 0; j < result[0].length; j++) {
                        System.out.print(result[i][j] + " ");
                    }
                    System.out.println();
                }
            }

            scanner.close();
        }

        public static int[][] addingMatrix(int[][] matrix1, int[][] matrix2) {
```



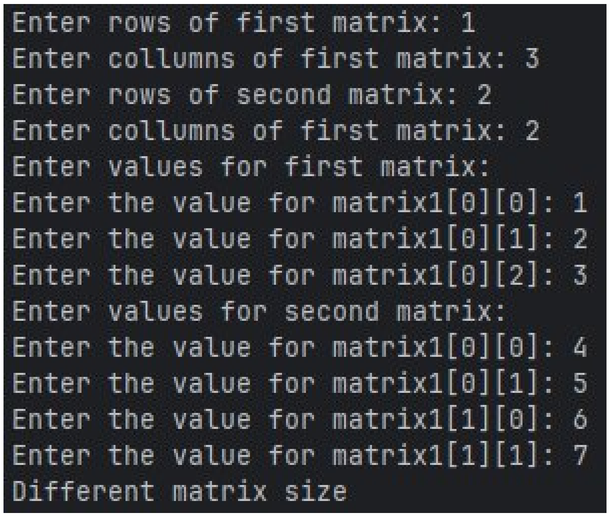
```
        if (matrix1.length == 0 || matrix2.length == 0) {
            System.out.println("Matrix is empty");
            return null;
        }

        int row1 = matrix1.length;
        int col1 = matrix1[0].length;
        int row2 = matrix2.length;
        int col2 = matrix2[0].length;

        if (row1 != row2 || col1 != col2) {
            System.out.println("Different matrix size");
            return null;
        }

        int[][] result = new int[row1][col1];

        for (int i = 0; i < row1; i++) {
            for (int j = 0; j < col1; j++) {
                result[i][j] = matrix1[i][j] + matrix2[i][j];
            }
        }

        return result;
    }
}
```

- ### Test 1: Normal case



```
Enter rows of first matrix: 2
Enter collumns of first matrix: 3
Enter rows of second matrix: 2
Enter collumns of first matrix: 3
Enter values for first matrix:
Enter the value for matrix1[0][0]: 1
Enter the value for matrix1[0][1]: 2
Enter the value for matrix1[0][2]: 3
Enter the value for matrix1[1][0]: 4
Enter the value for matrix1[1][1]: 5
Enter the value for matrix1[1][2]: 6
Enter values for second matrix:
Enter the value for matrix1[0][0]: 6
Enter the value for matrix1[0][1]: 5
Enter the value for matrix1[0][2]: 4
Enter the value for matrix1[1][0]: 3
Enter the value for matrix1[1][1]: 2
Enter the value for matrix1[1][2]: 1
Result matrix:
7 7 7
7 7 7
```

- Test 2: Different matrix size



```
Enter rows of first matrix: 1
Enter collumns of first matrix: 3
Enter rows of second matrix: 2
Enter collumns of first matrix: 2
Enter values for first matrix:
Enter the value for matrix1[0][0]: 1
Enter the value for matrix1[0][1]: 2
Enter the value for matrix1[0][2]: 3
Enter values for second matrix:
Enter the value for matrix1[0][0]: 4
Enter the value for matrix1[0][1]: 5
Enter the value for matrix1[1][0]: 6
Enter the value for matrix1[1][1]: 7
Different matrix size
```