

Lab 1. Environment Setup & Java Basics

IT3100E – 757054 – Object-oriented Programming

Họ và tên: Trần Việt Cường

MSSV:202416784

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

Notes

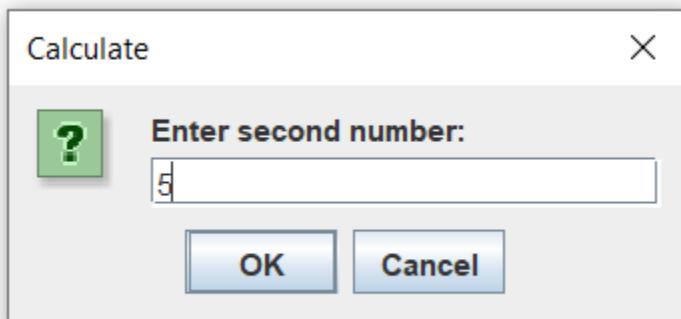
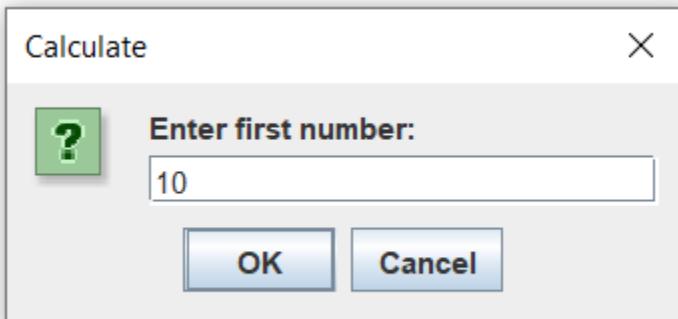
- To convert from String to double, you can use

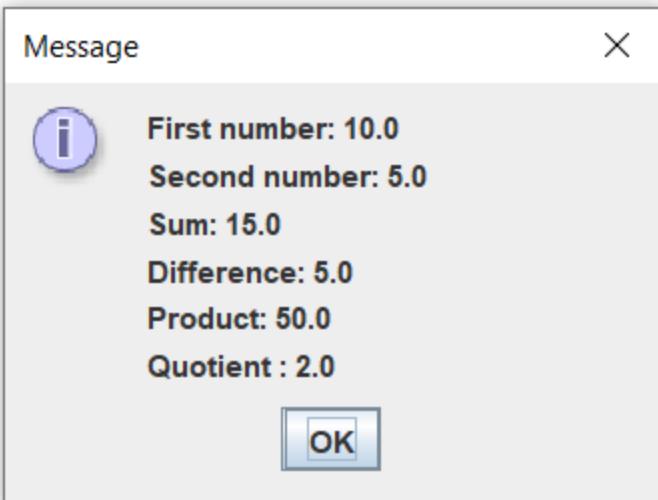
```
double num1 = Double.parseDouble(strNum1)
```

- Check the divisor of the division

Run

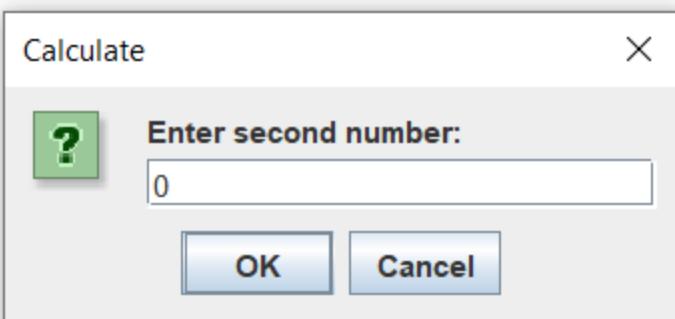
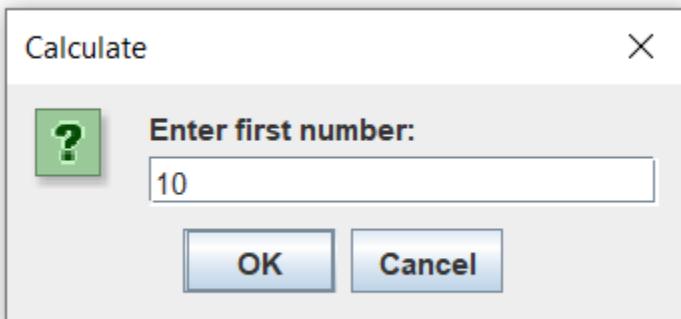
- Case the second number different from 0:

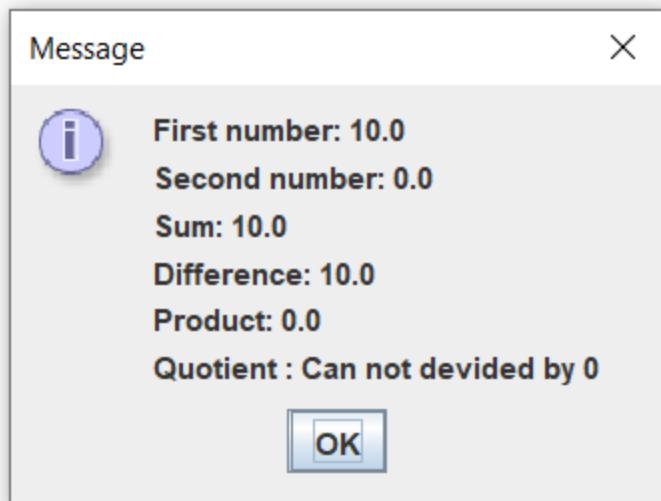




The program is running good

- Case the second number equal to 0:





The program can not devided by 0 and print “Can not devided by 0”

Exercise 2.2.6: Write a program to solve:

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

- **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.

- **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.

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

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:

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

- **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:

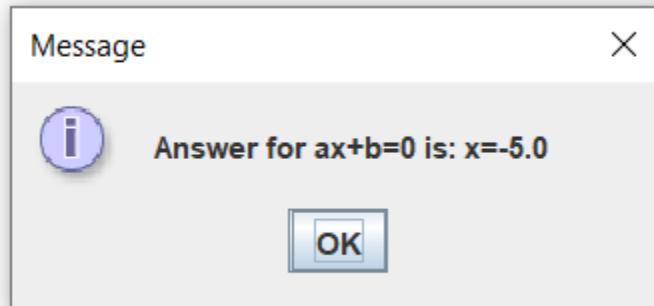
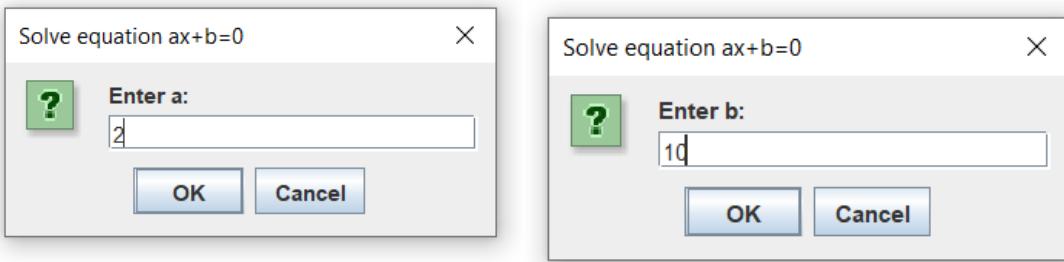
Use the discriminant $\Delta = b^2 - 4ac$

Run

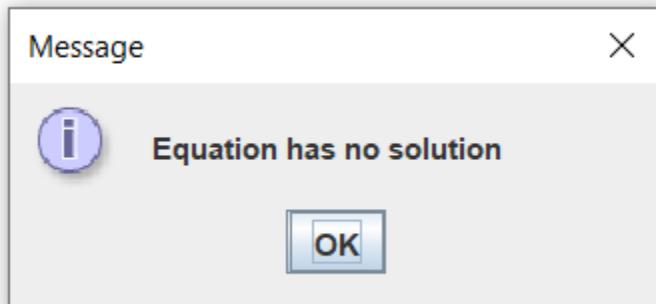
- **The first-degree equation (linear equation) with one variable:**

- With case both a and b different from 0

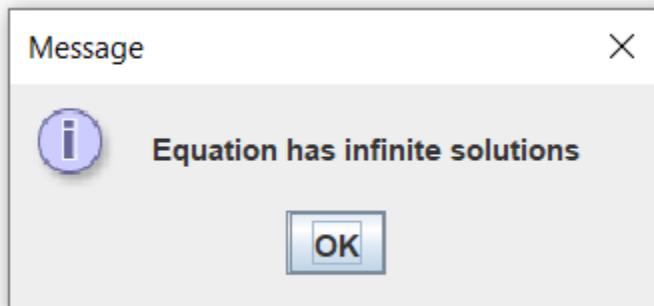
We will take an example as $2x+10 = 0$ and the program will handle the other.



- With case $a = 0$ and b different from 0:

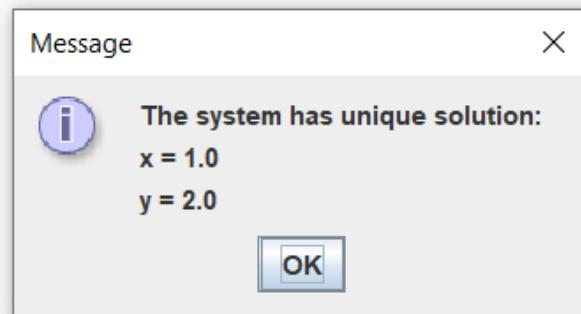
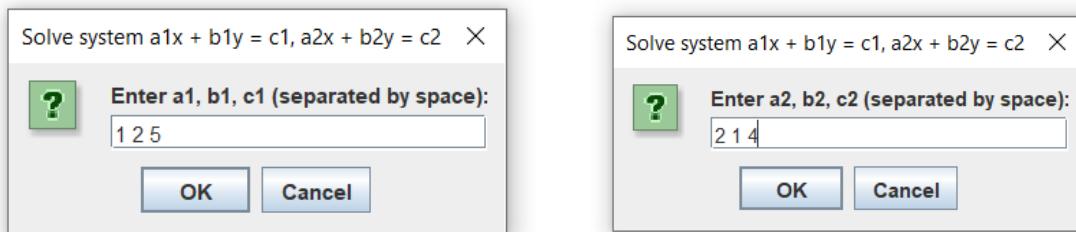


- With case $a = 0$ and $b = 0$:

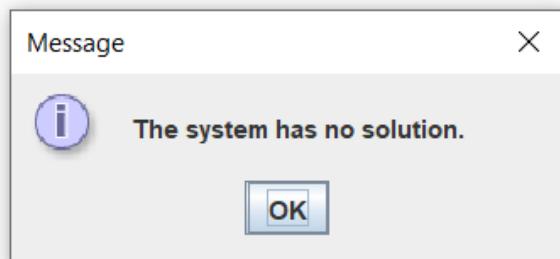


- **The system of first-degree equations (linear system) with two variables**
 - With case the system has unique root:

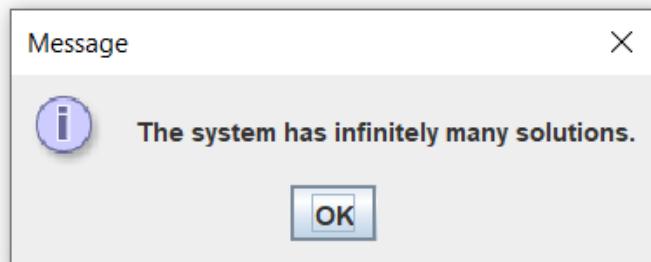
Let take an example of system: $x + 2y = 5$, $2x + y = 4$



- Case system has no root (eg: $x+y=1$, $x+y=2$):



- Case system has infinite roots (eg: $x+y=1$, $2x+2y=2$):



- The second-degree equation with one variable:

- Case equation has two distinct roots (eg: $x^2 + 2x - 3 = 0$):

The image shows four sequential dialog boxes from a software application titled "Solve equation ax^2+bx+c=0".

- The first box asks for "Enter a (different from 0)": The value "1" is entered.
- The second box asks for "Enter b": The value "2" is entered.
- The third box asks for "Enter c": The value "-3" is entered.
- The fourth box displays the results: "Equation has two distinct roots: x1 = 1.0, x2 = -3.0".

- Case equation has double roots (eg: $x^2 + 2x + 1 = 0$):

A dialog box titled "Solve equation ax^2+bx+c=0" displays the message: "Equation has double root: x1 = x2 = -1.0".

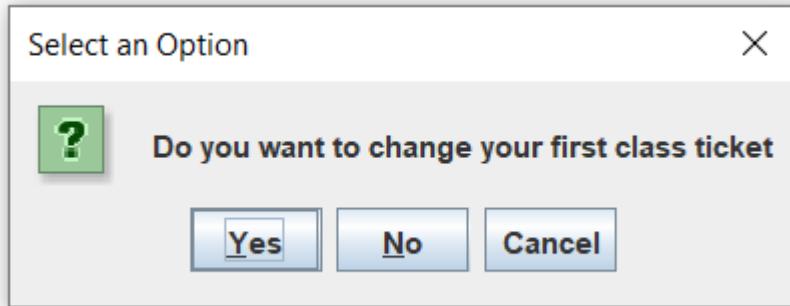
- Case equation has no root (eg: $x^2 + 2x + 3 = 0$):

A dialog box titled "Solve equation ax^2+bx+c=0" displays the message: "Equation has no root".

Exercise 6.1: Write, compile and run the ChoosingOption program:

Note: We use JavaBasics project for this exercise.

Run



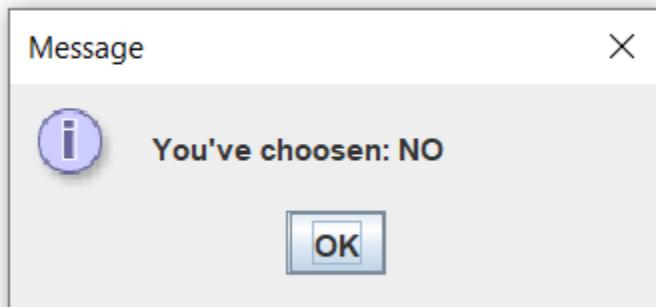
Answer the questions:

- *What happens if users choose “Cancel”?*

In program, the method JOptionPane.showConfirmDialog() returns an integer value depending on which button the user selects.

- Yes return JOptionPane.YES_OPTION (0)
- No return JOptionPane.NO_OPTION (1)
- Cancel return JOptionPane.CANCEL_OPTION (2)
- Close return JOptionPane.CLOSED_OPTION (-1)

But in the program, the code only checks JOptionPane.YES_OPTION and considers other options as No, then if the user chooses Cancel, the program will display “You’ve chosen: No”



- ***How to customize the options to users, e.g. only two options: “Yes” and “No”, OR “I do” and “I don’t” (Suggestion: Use Javadocs or using Eclipse/Netbean IDE help).***

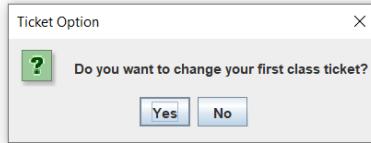
To customize the button labels, you can use JOptionPane.showOptionDialog() which allows you to specify your own text for the buttons.

- Example 1: “Yes” and “No” only (code and run)

```
package lab01;

import javax.swing.JOptionPane;
public class Bai61ChoosingOption {

    public static void main(String[] args) {
        String[] options = {"Yes", "No"};
        int option = JOptionPane.showOptionDialog(
            null,
            "Do you want to change your first class ticket?",
            "Ticket Option",
            JOptionPane.DEFAULT_OPTION,
            JOptionPane.QUESTION_MESSAGE,
            null,
            options,
            options[0]
        );
        String message;
        if (option == 0)
            message = "You've chosen: YES";
        else if (option == 1)
            message = "You've chosen: NO";
        else
            message = "Dialog closed without choosing.";
        JOptionPane.showMessageDialog(null, message);
        System.exit(0);
    }
}
```



- Example 2: “I do” and “I don’t”

```
package lab01;

import javax.swing.JOptionPane;
public class Bai61ChoosingOption {

    public static void main(String[] args) {
        String[] options = {"I do", "I don't"};
        int option = JOptionPane.showOptionDialog(
            null,
            "Do you ....?",
            "Ticket Option",
            JOptionPane.DEFAULT_OPTION,
            JOptionPane.INFORMATION_MESSAGE,
            null,
            options,
            options[0]
        );
        String message;
        if (option == 0)
            message = "You've chosen: I DO";
        else if (option == 1)
            message = "You've chosen: I DON'T";
        else
            message = "Dialog closed without choosing.";
        JOptionPane.showMessageDialog(null, message);
        System.exit(0);
    }
}
```



Exercise 6.2: Write a program for input/output from keyboard

Note: We use the JavaBasics project for this exercise.

Run



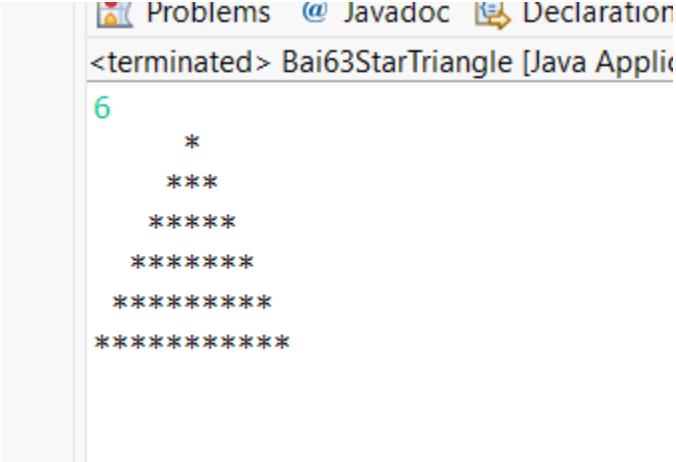
The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The title bar indicates the project is 'Bai62InputFromKeyboard [Java Application]' located at 'C:\Users\MrTran\p2\pool\plugins\org.e'. The console output is as follows:

```
<terminated> Bai62InputFromKeyboard [Java Application] C:\Users\MrTran\p2\pool\plugins\org.e
What's your name?
Tran Viet Cuong
How old are you?
19
How tall are you (m)?
1.75
Mr./Ms. Tran Viet Cuong, 19 years old. Your height is 1.75.
```

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

Example: n = 6

Run



The screenshot shows a Java application window with tabs for Problems, Javadoc, and Declaration. The title bar says '<terminated> Bai63StarTriangle [Java Application]'. The code area contains the number '6' in green, followed by a series of asterisks forming a triangle:

```
6
*
 ***
 *****
 ******
 *****
*****
```

Exercise 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.

Note: You must create a new Java project for this exercise.

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.

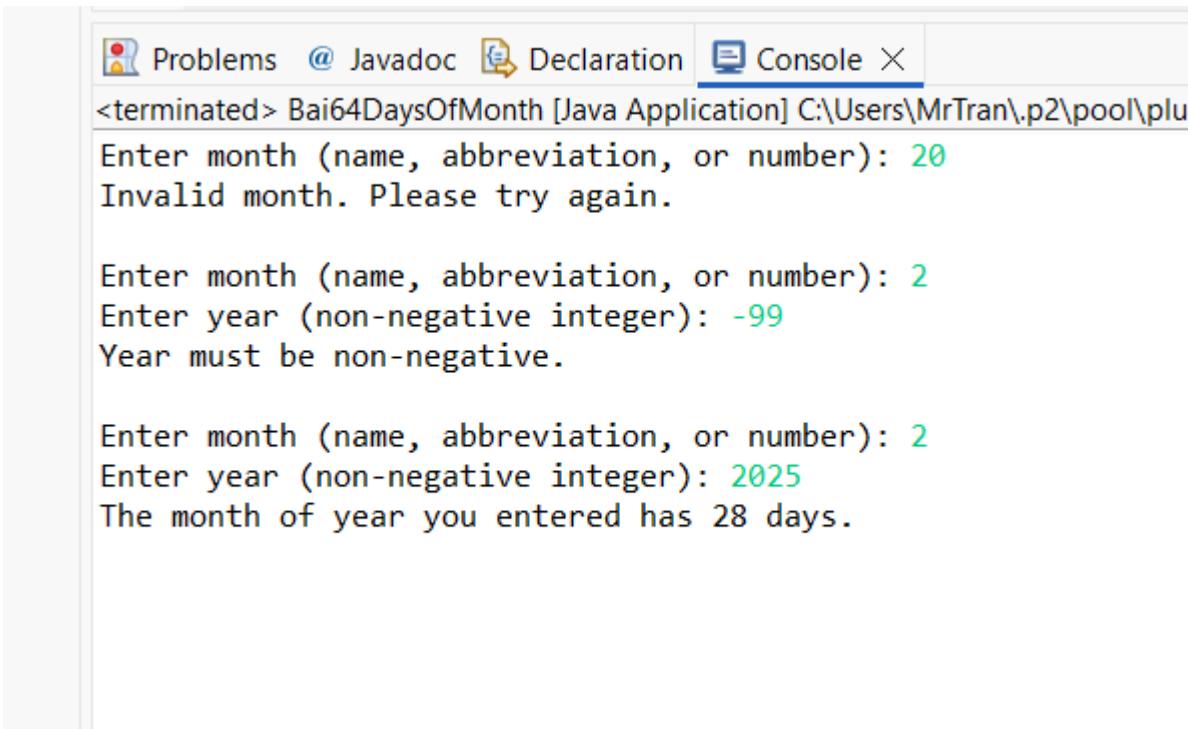
The user must enter a year in a non-negative number and enter all the digits. For instance, the valid inputs of year 1999 is only 1999, but not 99, “one thousand nine hundred ninety-nine”, or anything else.

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, year 1800 is not a leap year, yet 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

Run

Case not a leap year (including wrong month and wrong year input for example):



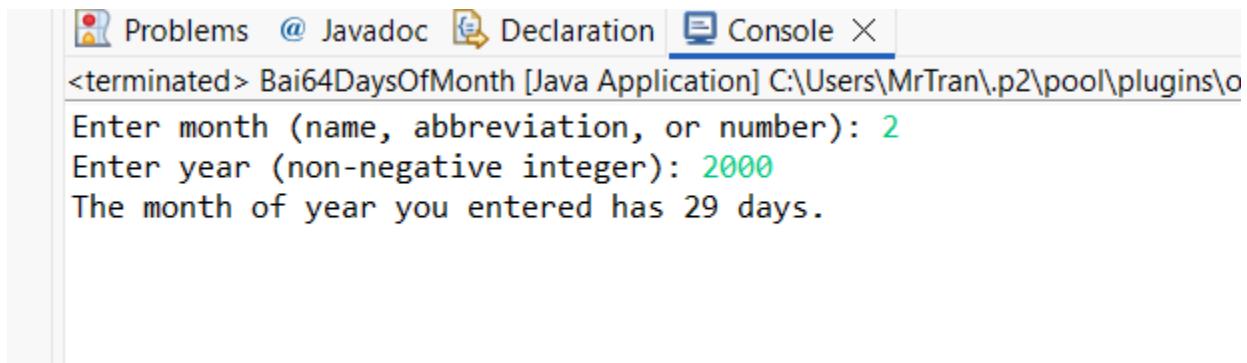
The screenshot shows the Eclipse IDE's Console view. The tabs at the top are Problems, Javadoc, Declaration, and Console (which is selected). The output window displays the following text:

```
<terminated> Bai64DaysOfMonth [Java Application] C:\Users\MrTran\p2\pool\plugins\o
Enter month (name, abbreviation, or number): 20
Invalid month. Please try again.

Enter month (name, abbreviation, or number): 2
Enter year (non-negative integer): -99
Year must be non-negative.

Enter month (name, abbreviation, or number): 2
Enter year (non-negative integer): 2025
The month of year you entered has 28 days.
```

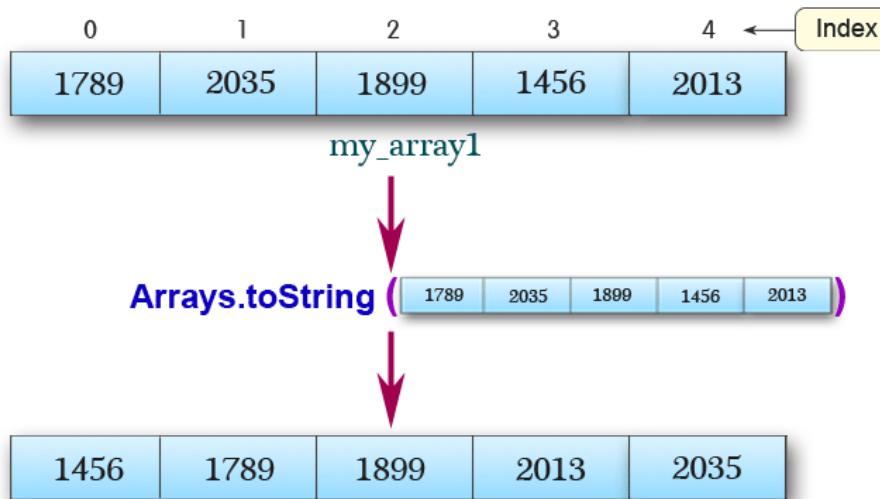
Case a leap year:



The screenshot shows the Eclipse IDE's Console view. The tabs at the top are Problems, Javadoc, Declaration, and Console (which is selected). The output window displays the following text:

```
<terminated> Bai64DaysOfMonth [Java Application] C:\Users\MrTran\p2\pool\plugins\o
Enter month (name, abbreviation, or number): 2
Enter year (non-negative integer): 2000
The month of year you entered has 29 days.
```

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



Note: You must create a new Java project for this exercise.

- The array can be entered by the user or a constant.

Run

```
Problems @ Javadoc Declaration Console X
<terminated> Bai65Array [Java Application] C:\Users\MrTran\.p2\pool\plugins\org.eclipse.justj.openjdk.h
Enter numbers (separated by spaces): 1789 2035 1899 1456 2013
Original array: [1789, 2035, 1899, 1456, 2013]
Sorted array: [1456, 1789, 1899, 2013, 2035]
Sum of the array is 9192
Average value of the array is 1838.4
```

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

Note: You must create a new Java project for this exercise.

- The matrices can be entered by the user or constants.

Run

```
<terminated> Bai66Matrices [Java Application] C:\Users\MrTran\p2\pool\plus
Enter number of rows: 3
Enter number of columns: 3
Enter Matrix A:
1 2 3
4 5 6
7 8 9
Enter Matrix B:
3 3 3
2 2 2
1 1 1
Sum of matrix A and matrix B is:
[4, 5, 6]
[6, 7, 8]
[8, 9, 10]
```