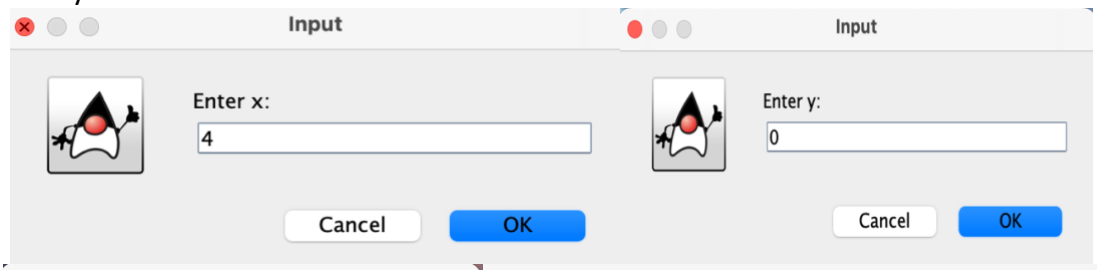
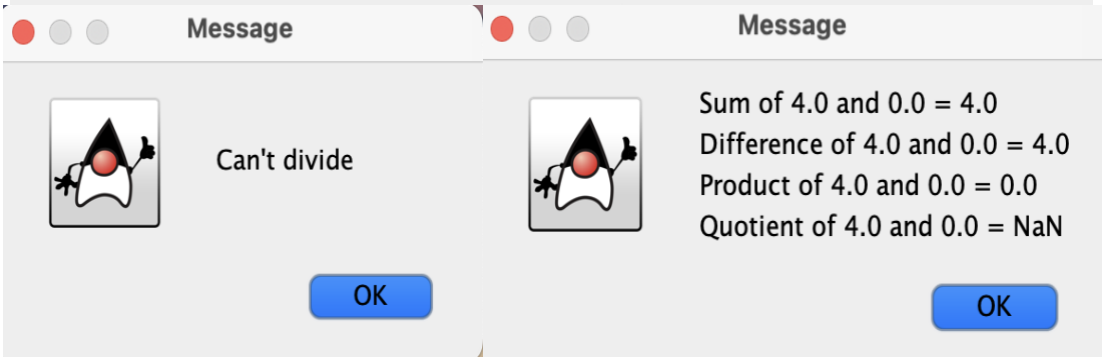
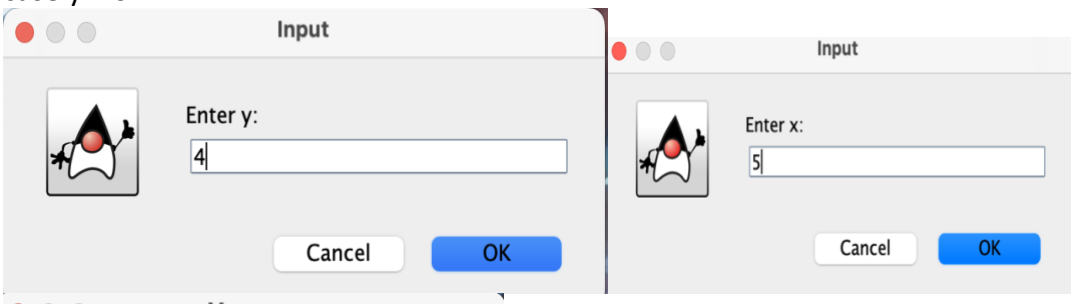
EX 2.2.5

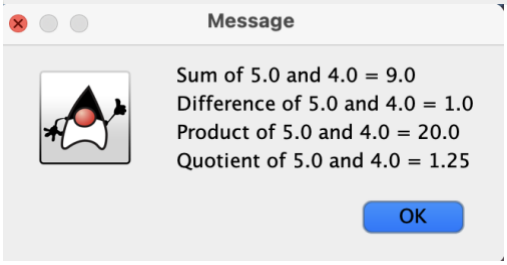
Case y =0




case y # 0

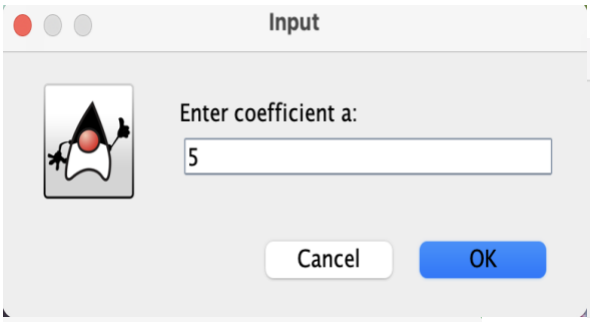


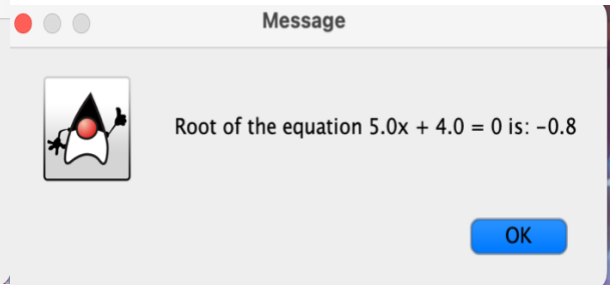



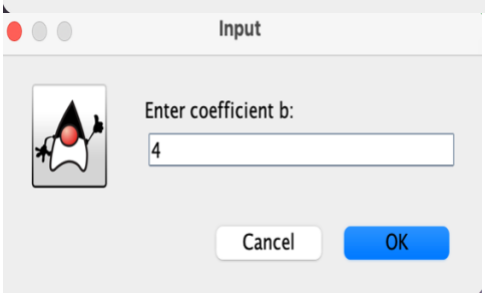
2.2.6

-Linear

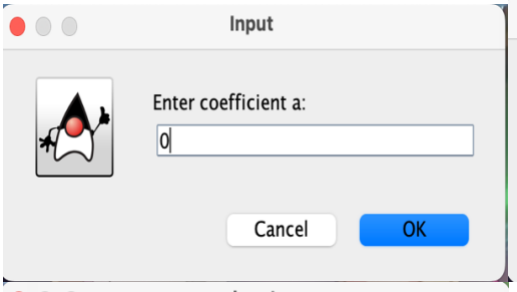
Case a#0

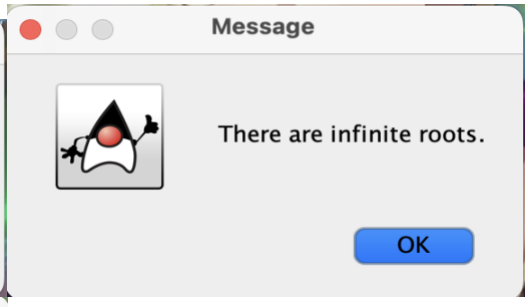


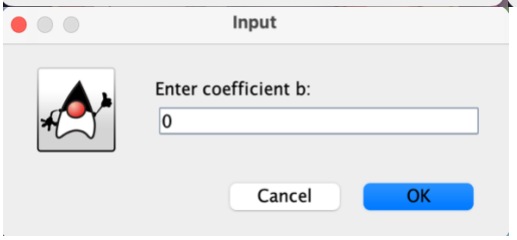



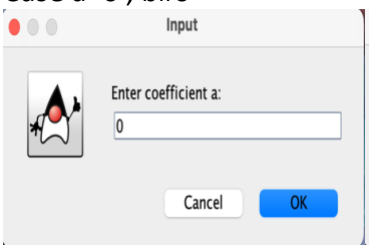

Case a =0 b=0

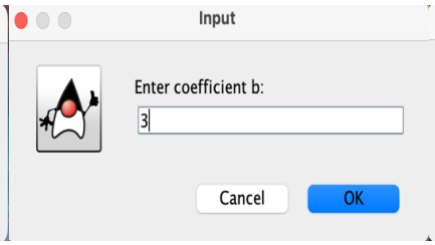



Case a=0 , b#0



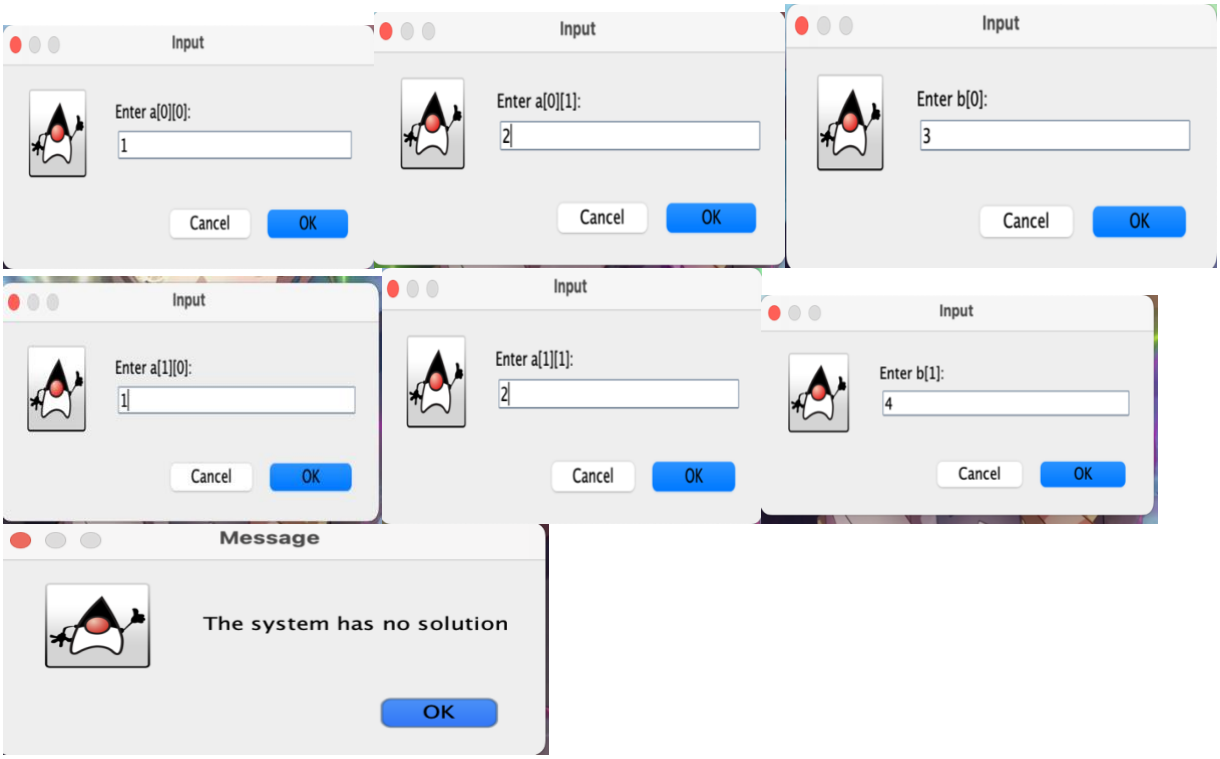




2.2.6

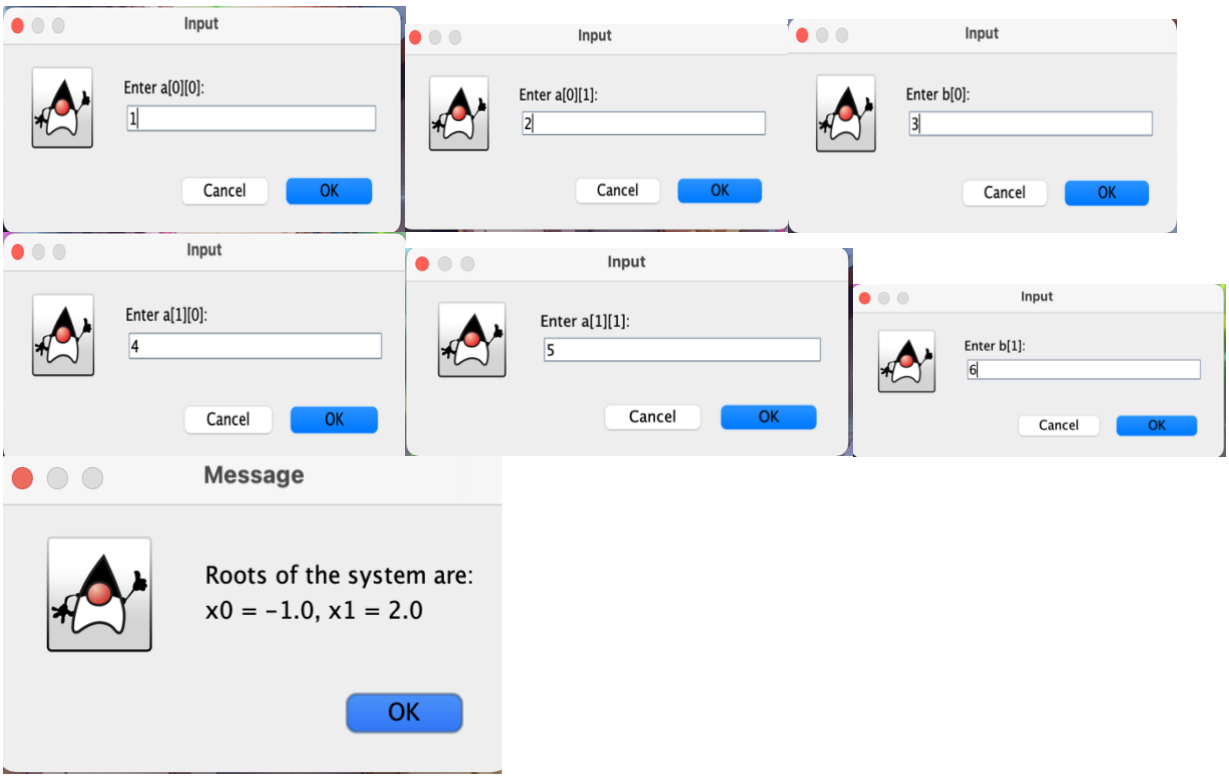
-Linear System

Case: infinite solution


Case: No solution



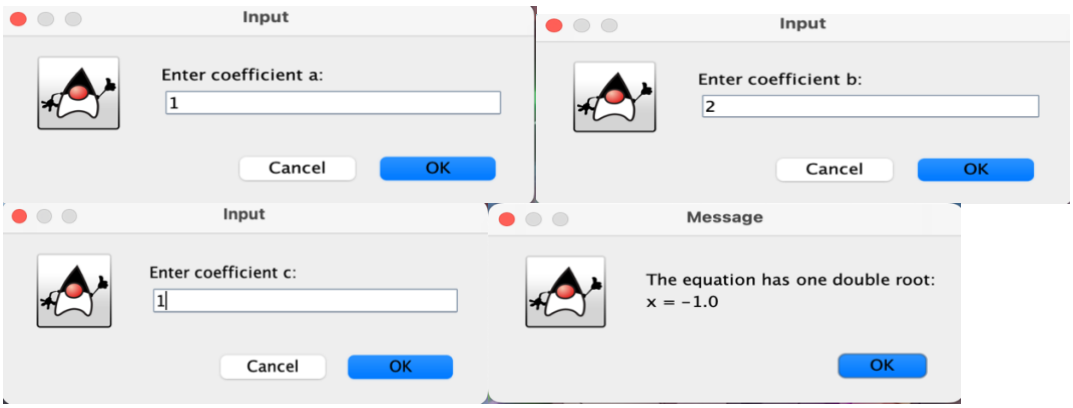
Case normal



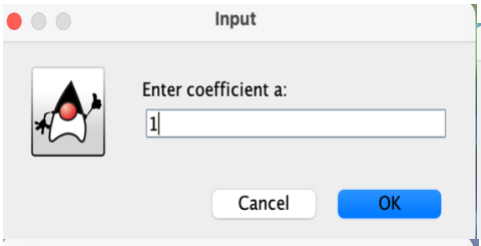

2.2.6

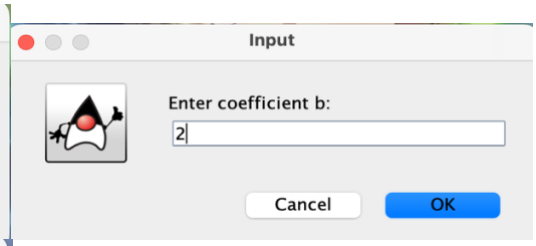
Second-degree

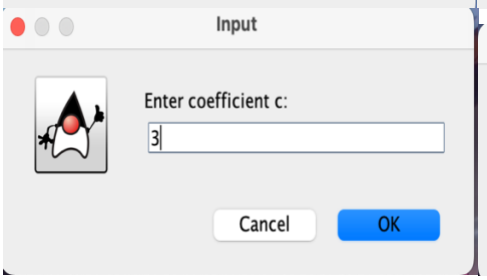
Case: Double root

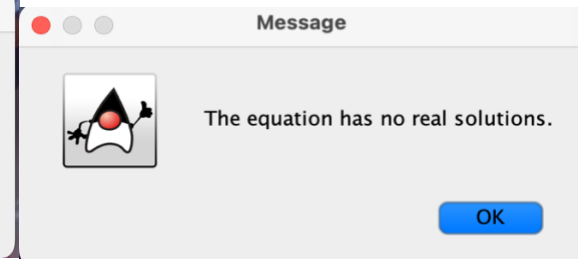


Case: No solution

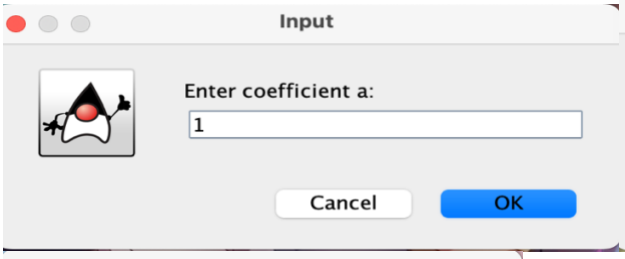


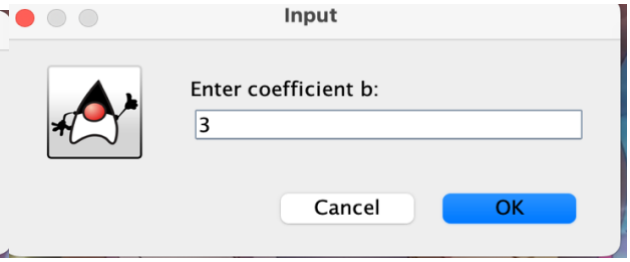



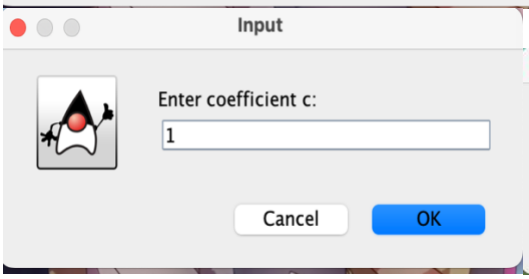



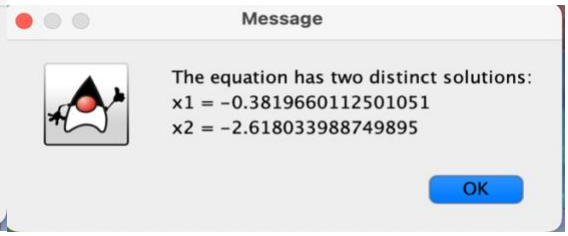




Case: Normal

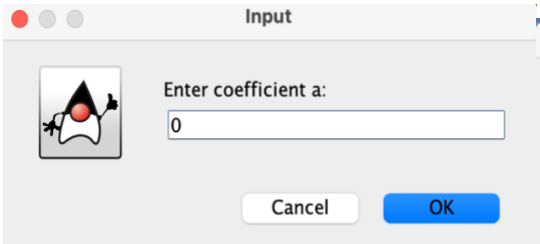


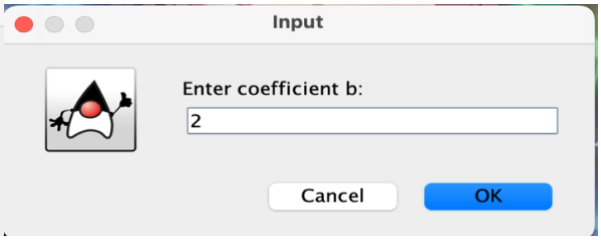



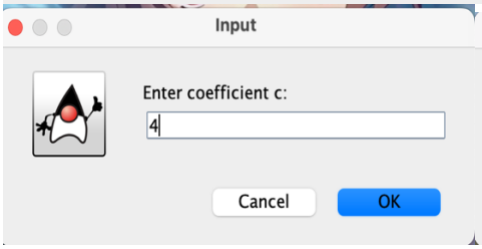



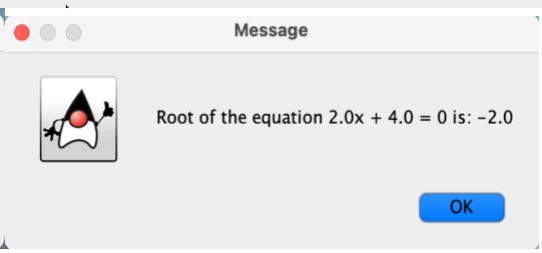




Case a=0 -> it will inherit linear model above

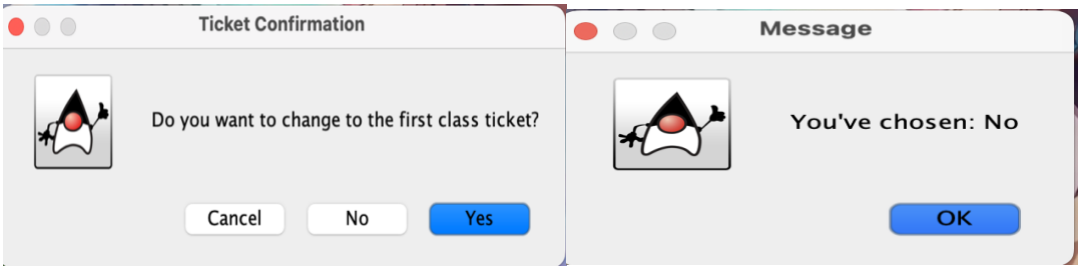


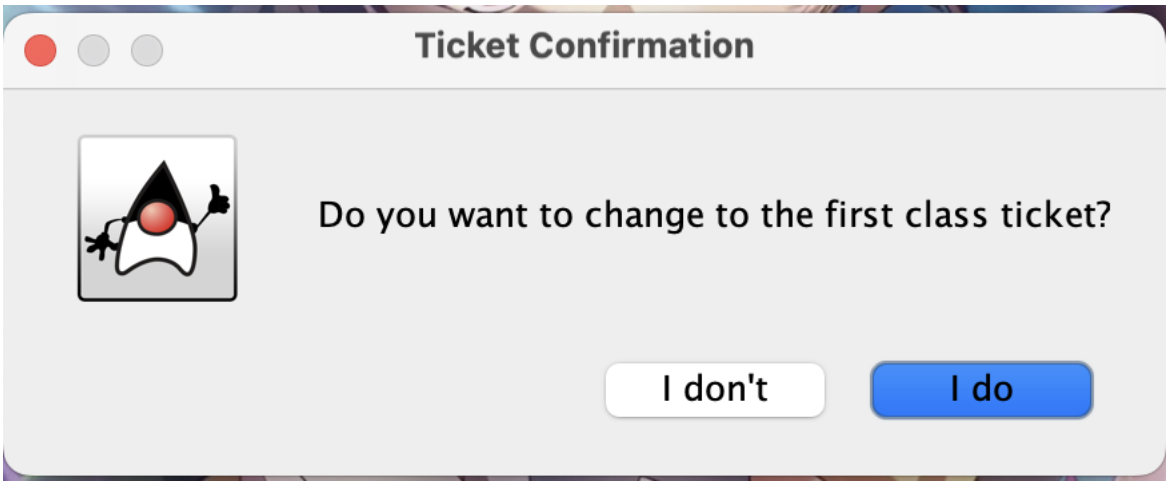



Exercise6.1




Customize




Answer question

Q1: If we choose cancel, I will give you No. Because Yes ~ 0, No ~1, Cancel~2

Therefore, when you choose cancel, option = 2.
And option(2) == Yes_Option(0). This statement is false
→ It returns No

```
                JOptionPane.YES_NO_CANCEL_OPTION
        );
    }

    public void run() {
        int option = showDialog();
        JOptionPane.showMessageDialog(parentComponent:null,
                "You've chosen: " + (option == JOptionPane.YES_OPTION ? "Yes" : "No"));
    }
```

Q2: We can customize button by doing from this

```
return JOptionPane.showConfirmDialog(
        parentComponent:null,
        message:"Do you want to change to the first class ticket?",
        title:"Ticket Confirmation",
        JOptionPane.YES_NO_CANCEL_OPTION
);
```

To this

```
Object[] options = {"I do", "I don't"};
return JOptionPane.showOptionDialog(
        parentComponent:null,
        message:"Do you want to change to the first class ticket?",
        title:"Ticket Confirmation",
        JOptionPane.YES_NO_OPTION,
        JOptionPane.QUESTION_MESSAGE,
        icon:null,
        options,
        options[0]);
```

The most difference between them is array options created by us, and we can design it as we want.

Exercise 6.2

```
(rl-for-topic-models) apple@Fushi exercise_6 % java exercise_6_2.java
What's your name ?
Nam
How old are you ?
18
How tall are you ?
1,65
 Mr/Ms Nam is 18. Your height is 1.65
(rl-for-topic-models) apple@Fushi exercise_6 %
```

Exercise 6.3

```
(rl-for-topic-models) apple@Fushi exercise_6 % java exercise_6_3.java
5
    *
   ***
  *****
 *******
*********
(rl-for-topic-models) apple@Fushi exercise_6 %
```

Exercise 6.4

```
(rl-for-topic-models) apple@Fushi exercise_6 % java exercise_6_4.java
Enter month: 0
Invalid month. Try again
Enter month: nam
Invalid month. Try again
Enter month: 2
Enter year: 2k
Invalid year. Try again
Enter year: 2100
The Month 2 of year 2100 has 28 days.
(rl-for-topic-models) apple@Fushi exercise_6 %
```

Exercise 6.5

```
(rl-for-topic-models) apple@Fushi exercise_6 % java exercise_6_5.java
Enter the number of elements
4
3 56 21 910
Sorted array:
3 21 56 910 %
(rl-for-topic-models) apple@Fushi exercise_6 %
```

Using quick_sort

```java
    static void swap(int a[], int i, int j){
    public static int partition(int left, int right, int a[]){
        int i = left;
        int j = right + 1;
        int pivot = a[left];
        while (true) {
            i = i +1;
            j = j -1;
            while (i <= right && a[i] < pivot){
                i++;
            }
            while (j > left && pivot < a[j]){
                j = j-1;
            }
            if (i > j) break;
                swap(a, i, j);
        }
        swap (a, left, j);
        return j;
    }
    public static void quick_sort (int left, int right, int a []){
        if (left < right){
            int pivot_index = partition(left, right, a);
            quick_sort(left, pivot_index -1 , a);
            quick_sort(pivot_index +1, right, a);
        }
    }
    Run | Debug
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println(x:"Enter the number of elements");
        int n = sc.nextInt();

        int a[] = new int[n];
        for (int i = 0; i < n; i++) {
            a[i] = sc.nextInt();
        }
        quick_sort(left:0, n-1, a);
        System.out.println(x:"Sorted array:");
        for (int i = 0; i < n; i++) {
            System.out.print(a[i] + " ");
        }
```

## Exercise 6.6

```
(rl-for-topic-models) apple@Fushi exercise_6 % java exercise_6_6.java
Enter size of column
3
Enter size of row
3
Enter matrix A
1 2 3
4 5 6
7 8 9
Enter matrix B
11 12 13
14 15 16
17 18 19

The final matrix:
12.0 14.0 16.0
18.0 20.0 22.0
24.0 26.0 28.0
```