Tutorial For Control Statement2

> Based on the tutorial of "2020S-Java-A" and "2020F-Java-A" designed by teaching group in SUSTech
>
> Modified (mainly change to markdown file) by ZHU Yueming
>
> Modified by JIA Yanhong in 2022. Sept. 18th
>
> Modified DaysOfYear by ZHU Yueming in 2024. Feb. 26th

Objective

2. Learn how to use the **for** repetition statement to execute statements in a program.
3. Learn how to use the **switch** selection statements to choose among alternative actions.
4. Learn how to use the **break and continue** statements in a program.

Exercise

Exercise 1

Calculate the value of π from the infinite series.

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \cdots$$

Input an integer **n** which represents the number of terms in the formula above. It is more precise when **n** is bigger. Use **for** statements to compute the value of π. Please using **double** format to define π.

Sample output:

```
Please input n:
10000
The estimational of Pi is 3.141493
```

hint:

The $increment_i$ of π would be `4.0 / (2 * i - 1)`

Exercise 2

Modify your program as follows:

Similar as the question above. Input a double value which represents a precision threshold. Your program should terminate when the difference between two successive iterations is smaller than the precision threshold. Print the value of π, and the iteration numbers. Please using **double** format to define π.

hint:

$Precision_i = increment_i$

Sample output:

```
Please input the precision:
0.0001
The estimational of Pi is 3.141643
It computed 20001 times
```

Think about this: when to use **for** and when to use **while**?

Exercise 3

> assignment from 2022 fall

You are given a list of numbers, representing the final-score of students of in your class, now you are required to count the distribution of grades.

Here the score is devided into 4 grades: A (90<=score<=100), B (80<=score<90), C (70<=score<80), D(60<=score<70).

Use **switch** to calculate the grade Level according to the following table.

Grade	Level
100~90	A
89~80	B
79~70	C
69~60	D
59~0	F

Input:

The user can input the credit and score of each course. The process should continue until the user inputs “-1”. Each line contains a single integer $\mathbf{S_i}$, representing a score. You are required to print the number of "A" , "B", "C", "D", "F" as required.

output:

Print **5** lines for each grade from grade A to grade E, following the format: "A: " + Number of A's.

Input sample:

```
94
92
88
65
90
77
50
-1
```

output sample:

```
A: 3
B: 1
C: 1
D: 1
F: 1
```

Exercise 4

DaysOfYear: There are 30 or 31 days in a month except February. There are 28 days in February in a common year, and 29 days in a leap year. Write a program to input year, month and day by Scanner and show the days of this year using **switch**.

A year is a leap year if it is:

(1) divisible by 4, but not divisible by 100;

(2) or divisible by 400;

According to the description above and the input and output. Please complete the sample code to finish the question.

Sample Code:

```
import java.util.Scanner;

public class DaysOfYear {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int year = in.nextInt();
        int month = in.nextInt();
        int day = in.nextInt();
        int daysOfYear = 0;
        boolean isLeapYear = _____;
```

```
        //calculate days before current month
          for (int i = 1; i < month; i++) {
              switch (i) {
                  ____________________
                      daysOfYear += 31;
                      break;
                  ____________________
                      daysOfYear += 30;
                      break;
                  ____________________
                      daysOfYear += (isLeapYear) ? 29 : 28;
              }
          }
        //add days in current month
          ____________________;
          System.out.println(daysOfYear);
    }
}
```

Sample inputs and Outputs:

```
> 2024 1 1
1
> 2024 2 1
32
> 2024 3 5
65
> 1 10 5
278
```

Exercise 5

(1) Print a multiplication table as follows:

```
1 * 1 =  1
1 * 2 =  2  2 * 2 =  4
1 * 3 =  3  2 * 3 =  6  3 * 3 =  9
1 * 4 =  4  2 * 4 =  8  3 * 4 = 12  4 * 4 = 16
1 * 5 =  5  2 * 5 = 10  3 * 5 = 15  4 * 5 = 20  5 * 5 = 25
1 * 6 =  6  2 * 6 = 12  3 * 6 = 18  4 * 6 = 24  5 * 6 = 30  6 * 6 = 36
1 * 7 =  7  2 * 7 = 14  3 * 7 = 21  4 * 7 = 28  5 * 7 = 35  6 * 7 = 42  7 * 7 =
49
1 * 8 =  8  2 * 8 = 16  3 * 8 = 24  4 * 8 = 32  5 * 8 = 40  6 * 8 = 48  7 * 8 =
56  8 * 8 = 64
1 * 9 =  9  2 * 9 = 18  3 * 9 = 27  4 * 9 = 36  5 * 9 = 45  6 * 9 = 54  7 * 9 =
63  8 * 9 = 72  9 * 9 = 81
```

Complete the code below to solve this question:

```
public static void main(String[] args) {
        for (int i = 1; i <= 9; i++) {
            for (int j = 1; _________________) {
                System.out.printf("%d * %d = %2d\t", ____________);
            }
            _________________
        }
    }
```

(2) Design a program can display a number of [1, 9].

- at the beginnning of each loop, print `Please input a number between [1,9]`
- If the input number is in [1, 9], display `number [your input] is in [1, 9]`, and then input a number again.
- If the input number is 0, terminate the program.
- If the input number is smaller than 0 or is larger than 9, display `Your number is not in [1,9]`

  Sample input and output:

  ```
  Please input a number between [1,9]:
  5
  number 5 is in [1, 9]
  Please input a number between [1,9]:
  10
  Your number is not in [1,9]
  Please input a number between [1,9]:
  2
  number 2 is in [1, 9]
  Please input a number between [1,9]:
  0
  ```

  Complete the code below to solve this question:

  ```
  public static void main(String[] args) {
          Scanner in = new Scanner(System.in);
          do {
              System.out.println("Please input a number between [1,9]:");
              int number = in.nextInt();
              if (___________________) {
                  System.out.printf("number %d is in [1, 9]\n", number);
              } else if (_________________) {
                  break;
              }else{
                  System.out.println("Your number is not in [1,9]");
  ```

```
            }
        } while (__________);
    }
```

Final Exercise 5 : Modify the program so that

- The program can display a multiplication table of any given size in `[1, 9]`.
- The program keeps running until the user inputs 0.
- The program will warn users for invalid inputs.
- Try to use `break` and `continue` statements to complete the task.

Sample output:

```
Please input a number between [1,9]:
3
1 * 1 =  1
1 * 2 =  2  2 * 2 =  4
1 * 3 =  3  2 * 3 =  6  3 * 3 =  9
Please input a number between [1,9]:
4
1 * 1 =  1
1 * 2 =  2  2 * 2 =  4
1 * 3 =  3  2 * 3 =  6  3 * 3 =  9
1 * 4 =  4  2 * 4 =  8  3 * 4 = 12  4 * 4 = 16
Please input a number between [1,9]:
10
Your number is not in [1,9]
Please input a number between [1,9]:
0
```