Lab: Nested Loops

Problems for exercise and homework for the "**Programming Basics**" course @ SoftUni Global.

Submit your solutions to the **SoftUni Judge** system at: https://judge.softuni.org/Contests/3697

1. Clock

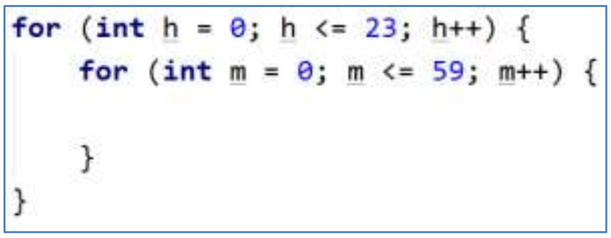
Write a program that prints the **hours of the day from 0:0 to 23:59**, each on a separate line. The hours must be displayed in the format **"{hour}:{minutes}"**.

Sample Input and Output

Input	Output
(no input)	0:0 0:1 0:2 0:3 0:4 0:5 0:6 0:7 0:8 0:9 0:10 ... 23:50 23:51 23:52 23:53 23:54 23:55 23:56 23:57 23:58 23:59

Hints and Guidelines

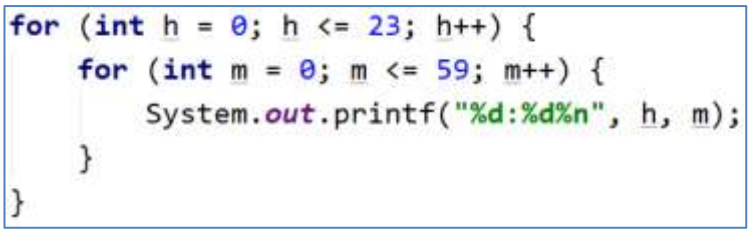
1. Create 2 nested for-loops to iterate every minute and hour of the day:



```
for (int h = 0; h <= 23; h++) {
    for (int m = 0; m <= 59; m++) {

    }
}
```

2. Print the result:



```
for (int h = 0; h <= 23; h++) {
    for (int m = 0; m <= 59; m++) {
        System.out.printf("%d:%d%n", h, m);
    }
}
```

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3697#0

2. Multiplication Table

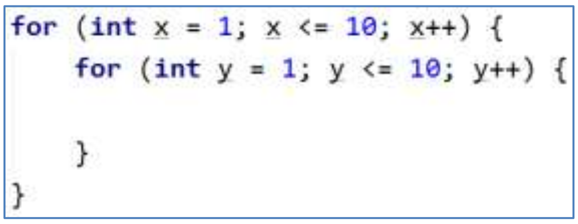
Print on the console the multiplication table for the numbers 1 to 10 in the format **"{first multiplier} * {second multiplier} = {result}"**.

Sample Input and Output

Input	Output
(no input)	1 * 1 = 1 1 * 2 = 2 1 * 3 = 3 1 * 4 = 4 1 * 5 = 5 1 * 6 = 6 1 * 7 = 7 1 * 8 = 8 1 * 9 = 9 1 * 10 = 10 ... 10 * 1 = 10 10 * 2 = 20 10 * 3 = 30 10 * 4 = 40 10 * 5 = 50 10 * 6 = 60 10 * 7 = 70 10 * 8 = 80 10 * 9 = 90 10 * 10 = 100

Hints and Guidelines

1. Add 2 nested for-loops to iterate every possible value of the two factors from 1 to 10:



```
for (int x = 1; x <= 10; x++) {
    for (int y = 1; y <= 10; y++) {

    }
}
```

2. Find the product of the two factors and print the result:

```java
for (int x = 1; x <= 10; x++) {
    for (int y = 1; y <= 10; y++) {
        int product = x * y;
        System.out.printf("%d * %d = %d%n",
                x, y, product);
    }
}
```

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3697#1

3. Combinations

Write a program that calculates **how many solutions in natural numbers** (including zero) has the equation:

x1 + x2 + x3 = n

The number n is an integer and is entered from the console.

Sample Input and Output

Input	Output	Comments	Input	Output	Input	Output
25	351	We generate all combinations of 3 numbers, the first being: 0+0+0=0, but since it is not equal to 25, we continue: 0+0+1=1 – also not 25, etc. We come to the first valid combination: 0 + 0 + 25 = 25, we increase the number of valid combinations by 1, the second valid combination is: 0 + 1 + 24 = 25 The third: 0 + 2 + 23 = 25, etc. After generating all possible combinations, the number of the valid one is 351.	20	231	5	21

Hints and Guidelines

1. Read the input from the console - **an integer:**

```java
Scanner scan = new Scanner(System.in);
int n = Integer.parseInt(scan.nextLine());
```

2. Create 3 nested **for-loops** to iterate every possible value of one of the 3 numbers in the equation:

```java
// x1 + x2 + x3 = n
for (int x1 = 0; x1 <= n; x1++) {
    for (int x2 = 0; x2 <= n; x2++) {
        for (int x3 = 0; x3 <= n; x3++) {

        }
    }
}
```

3. Check the innermost nested loop for the values of **x1**, **x2**, and **x3** in each iteration. For the equation to be valid, its sum must be equal to **n**. Create a **validCombinationsCount** variable to store the number of valid combinations and add to it each time you generate one:

```java
int validCombinationsCount = 0;
```

```java
for (int x1 = 0; x1 <= n; x1++) {
    for (int x2 = 0; x2 <= n; x2++) {
        for (int x3 = 0; x3 <= n; x3++) {
            // Increment counter
        }
    }
}
```

4. Finally, print the number of valid combinations (`validCombinationsCount`).

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3697#2

4. Sum of Two Numbers

Write a program that checks **all possible combinations of a pair of numbers in the interval of two given numbers**. The output is printed, **which is the combination** whose **sum of numbers is equal** to a given **magic number**. If **no combination** matches the condition, **a message is printed that it was not found**.

Input Data

3 lines are read from the console:

- **First line - beginning of the interval** - **an integer in the range [1...999]**
- **Second line - end of the interval** - **an integer in the range [greater than the previous...1000]**
- **Third line - a magic number - an integer in the range [1...10000]**

Output Data

One line is printed on the console:

- If a combination **is found** whose sum of numbers is **equal to the magic number**:
  - `"Combination N:`**{sequence number}** `(`**{first number}** `+` **{second number}** `=` **{magic number}**`)"`
- If **no matching** condition is found:
  - `"`**{number of all combinations}**` combinations - neither equals `**{magic number}**`"`

Sample Input and Output

Input	Output	Comments	Input	Output

`1`
`10`
`5`
	`Combination N:4 (1 + 4 = 5)`	All combinations of two numbers between 1 and 10 are:
		1 1, 1 2, 1 3, **1 4**, 1 5, ... 2 1, 2 2, ... 4 9, 4 10, 5 1 ... 10 9, 10 10
		The first combination whose sum of numbers is equal to the magic number 5 is **the fourth (1 and 4)**
			`88`
			`888`
			`1000`
				`Combination N:20025 (112 + 888 = 1000)`

Input	**Output**	**Comments**	**Input**	**Output**

`23`
`24`
`20`
	`4 combinations - neither equals 20`	All combinations of two numbers between 23 and 24 are: 23 23, 23 24, 24 23, 24 24 (total 4)
		There are no pairs of numbers whose sum is equal to the magic 20
			`88`
			`888`
			`2000`
				`641601 combinations - neither equals 2000`

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3697#3

5. Travelling

Sophie loves to travel and wants to visit **several different destinations** this year. When she chooses a destination, she will decide **how much money she will need** to get there and start **saving**. When she has saved enough, she will be able to travel.

On the console, each time a destination and a minimum budget needed for the trip are read from the console.

Then a few sums will be read, **floating-point numbers** that Sophie saves by working, and when **she manages** to raise enough for the trip, she will leave, and the console should print:

```
"Going to {destination}!"
```

When she has visited all the destinations she wants, instead of a destination she will enter "**End**" and the program will end.

Sample Input and Output

Input	Output	Input	Output

Greece 1000 200 200 300 100 150 240 Spain 1200 300 500 193 423 End	Going to Greece! Going to Spain!	France 2000 300 300 200 400 190 258 360 Portugal 1450 400 400 200 300 300 Egypt 1900 1000 280 300 500 End	Going to France! Going to Portugal! Going to Egypt!

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3697#4

6. Building

Write a program that displays the numbers of the rooms in a building (in **descending** order) on the console, provided that the following conditions are met:

- On each **even floor,** there are **offices**
- On each **odd floor,** there are **apartments**
- Each **apartment** is marked as follows: **`"A{floor number}{apartment number}"`**, apartment numbers start from 0
- Each office is named: **`"O{floor number}{apartment number} "`**, office numbers also start from 0

There are always apartments on the top floor, and they are bigger than the others, so their number says '**L**' instead of '**A**'. If there is only one floor, there are only large apartments!
Two integers are read from the console - **the number of floors and the number of rooms per floor**.

Sample Input and Output

Input	Output	Comments

6
4
	`L60 L61 L62 L63`
	`A50 A51 A52 A53`
	`O40 O41 O42 O43`
	`A30 A31 A32 A33`
	`O20 O21 O22 O23`
	`A10 A11 A12 A13`
		We have a total of 6 floors, with 4 rooms per floor. The odd floors have only apartments and even only offices.

Input	Output	Input	Output

9 5	L90 L91 L92 L93 L94 O80 O81 O82 O83 O84 A70 A71 A72 A73 A74 O60 O61 O62 O63 O64 A50 A51 A52 A53 A54 O40 O41 O42 O43 O44 A30 A31 A32 A33 A34 O20 O21 O22 O23 O24 A10 A11 A12 A13 A14	4 4	L40 L41 L42 L43 A30 A31 A32 A33 O20 O21 O22 O23 A10 A11 A12 A13

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3697#5