

# Methods - Exercises

1.	Sign of Integer	1
2.	Grades	1
3.	Printing Triangle	2
4.	Calculator	2
5.	Shop	2
6.	Calculate Rectangle Area	3
7.	Repeat String	3
8.	Math Power	3
9.	Greater of Two Values	3
10.	Multiply Evens by Odds	4
11.	Math Operations	4
12.	Password Validator	5

## 1. Sign of Integer

Create a method that prints the sign of an integer number.

Input	Output
2	The number 2 is positive.
-5	The number -5 is negative.
0	The number 0 is zero.

## 2. Grades

Write a method that **receives a grade** between **2.00** and **6.00** and **prints the corresponding grade in words**:

- 2.00 – 2.99 - "**Fail**"
- 3.00 – 3.49 - "**Poor**"
- 3.50 – 4.49 - "**Good**"
- 4.50 – 5.49 - "**Very good**"
- 5.50 – 6.00 - "**Excellent**"

Input	Output
3.33	Poor



<b>4.50</b>	Very good
<b>2.99</b>	Fail

### 3. Printing Triangle

Create a method for printing triangles as shown below:

Input	Output
<b>3</b>	1 1 2 1 2 3 1 2 1
<b>4</b>	1 1 2 1 2 3 1 2 3 4 1 2 3 1 2 1

### 4. Calculator

Write a program that receives a string on the first line ("**add**", "**multiply**", "**subtract**", "**divide**") and on the next two lines receives **two** numbers. Create four methods (for each calculation) and invoke the right one depending on the command. The method should also print the result (it needs to be void).

Input	Output
<b>subtract</b> <b>5</b> <b>4</b>	1
<b>divide</b> <b>8</b> <b>4</b>	2

### 5. Shop

Write a method that calculates the total price of an order and prints it on the console. The method should receive **one of the following products**: coffee, water, coke, snacks, and a **quantity** of the product. The prices for a single piece of each product are:

- **coffee** – 1.50
- **water** – 1.00
- **coke** – 1.40
- **snacks** – 2.00



Print the result rounded to the **second** decimal place.

Input	Output
water 5	5.00
coffee 2	3.00

## 6. Calculate Rectangle Area

Create a method that calculates and **returns** the area of a rectangle by given width and length.

Input	Output
3 4	12
5 7	35

## 7. Repeat String

Write a method that receives a string and a repeat count **n** (integer). The method should return a new string (the old one repeated n times).

Input	Output
abc 3	abcabcabc
String 2	StringString

## 8. Math Power

Create a method that calculates and returns the value of a number raised to a given power.

### Examples

Input	Output
2 8	256
5.5 3	166.37 5

Create a method that will have two parameters - the number and the power, and will return a result of type **double**.



## 9. Greater of Two Values

You are given two values of the same type as input. The values can be of type **int**, **char** or **String**. Create a method **getMax()** that returns the greater of the two values:

Input	Output
int 2 16	16
char a z	z
string Ivan Todor	Todor

1. You need to create three methods with the same name and different signatures for this method.
2. Create a method that will compare integers.
3. Create a second method with the same name, which will compare characters.
4. Lastly, you need to create a method to compare strings. This is a bit different as strings don't allow to be compared with the operators ">" and "<".

You need to use the method "**compareTo()**", which returns an integer value (greater than zero if the compared object is greater, less than zero if the compared object is lesser, and zero if the two objects are equal).

5. The last step is to read the input. Use appropriate variables and call the **getMax()** from your **main()**.

## 10. Multiply Evens by Odds

Create a program that reads an **integer** number and **multiplies the sum of all its even digits by the sum of all its odd digits**:

Input	Output
12345	54
12345	54



## 11. Math Operations

Write a method that receives **two numbers** and **an operator**, calculates the result, and returns it. You will be given three lines of input. The first will be the first number, the second one will be the operator, and the last one will be the second number. The possible operators are: **/ \* + -**

Print the result rounded up to the **second** decimal point.

### Example

Input	Output
5 * 5	25
4 + 8	12

## 12. Password Validator

Write a program that validates passwords. Password rules are:

- 6 – 10 characters (**inclusive**)
- Consists only of **letters** and **digits**.
- Have at least **2** digits.

If a password is valid, print **"Password is valid"**. If it is not valid, for every unfulfilled rule, print a message:

- **"Password must be between 6 and 10 characters"**.
- **"Password must contain only letters and digits"**.
- **"Password must have at least 2 digits"**.

### Examples

Input	Output
logIn	Password must be between 6 and 10 characters Password must have at least 2 digits
MyPass12	Password is valid



<b>3</b>	
<b>PaSSS.</b>	Password must contain only letters and digits Password must have at least 2 digits

