

Exercises: Functions, Debugging and Troubleshooting Code

Problems for exercises and homework for the [“Python Fundamentals” course @ SoftUni](#). Submit your solutions in the SoftUni judge system at <https://judge.softuni.bg/Contests/922/Functions-and-Debugging>.

I. Declaring and Invoking Functions

01. Blank Receipt

Create a function that prints a blank cash receipt. The function should invoke three other functions: one for printing the header, one for the body and one for the footer of the receipt.

The header should contain the following text:	CASH RECEIPT -----
The body should contain the following text:	Charged to_____ Received by_____
And the text for the footer:	----- © SoftUni

Examples

Output
CASH RECEIPT ----- Charged to_____ Received by_____
----- © SoftUni

Hints

1. First create a function with no parameters for printing the header. Give it a **meaningful name** and write the code that it will execute.
2. Do the same for printing the receipt body and footer.
3. Create a **function that will call all three functions** in the necessary order. Again, give it a **meaningful and descriptive name** and write the code.
4. For printing "©" use Unicode "\u00A9"
5. **Call** (invoke) the first function

02. Sign of Integer Number

Create a function that prints the sign of an integer number n.

Examples

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

Hints

1. Create a function with a **descriptive name** which should receive **one parameter**.
2. Implement the body of the function by handling different cases:
 - a. If the number is greater than zero
 - b. If the number is less than zero
 - c. And if the number is equal to zero
3. Call (invoke) the newly created function.

03. Printing Triangle

Create a function for printing triangles as shown below:

Examples

Input	Output
3	1 1 2 1 2 3 1 2 1
4	1 1 2 1 2 3 1 2 3 4 1 2 3 1 2 1

Hints

1. After you read the input.
2. Start by creating a function **for printing a single line** from a **given start** to a **given end**. Choose a **meaningful name** for it, describing its purpose.
3. Think how you can use it to solve the problem.
4. After you spent some time thinking, you should have come to the conclusion that you will need two loops
5. In the first loop you can print the first half of the triangle without the middle line.
6. Next, print the middle line.
7. Lastly, print the rest of the triangle.

04. Draw a Filled Square

Draw at the console a filled square of size n like in the example:

Examples

Input	Output
4	----- -\\\\\\/- -\\\\\\/- -----

Hints

1. Read the input
2. Create a function which will print the top and the bottom rows (they are the same). Don't forget to give it a descriptive name and to give it as a parameter some length
3. Create the function which will print the middle rows.
4. Use the functions that you've just created to draw a square.

II. Returning Values and Overloading

05. Calculate Triangle Area

Create a function that calculates and **returns** the [area](#) of a triangle by given base and height.

Use a **general formatting** with **12 digits** after the decimal point (e.g. `{area: .12g}`)

Examples

Input	Output
3 4	6

Hints

1. After reading the input
2. Create a function that calculates the area.
3. **Invoke** the function in the main and **save the return value in a new variable**.

06. Math Power

Create a function that calculates and returns the value of a number raised to a given power:

Examples

Input	Output
2 8	256.0
1.5 3	3.375

Hints

1. As usual, read the input
2. Create a function which will have two parameters - the number and the power, and will return a result
3. Print the result with no specific formatting.

07. 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 function that returns the greater of the two values:



Examples

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

Hints

1. For this function you need to create three functions with the same name and different signatures
2. Create a function which will compare integers.
3. Lastly you need to create a function to compare the other types.
4. The last step is to read the input, use appropriate variables and call the function you've just written.

III. Debugging and Program Flow

08. 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**:

Examples

Input	Output	Comments
12345	54	12345 has 2 even digits - 2 and 4. Even digits has sum of 6 . Also it has 3 odd digits - 1, 3 and 5. Odd digits has sum of 9 . Multiply 6 by 9 and you get 54 .
-12345	54	

Hints

1. Create a function with a **name describing its purpose**. The function should have a **single parameter** and will return value. Also the function will call two other functions.
2. Create two other functions each of which will sum either even or odd digits.
3. Implement the logic for summing odd digits.
4. Do the same for the function that will sum even digits.
5. As you test your solution you may notice that it doesn't work for negative numbers. Following the program execution line by line, find and fix the bug.