# Lab: Error Handling

Problems for in-class lab for the[**Python Advanced Course @SoftUni**](https://softuni.bg/trainings/3963/python-advanced-january-2023)**.**

## So Many Exceptions

You are provided with the following code. This code raises many exceptions. Fix it, so it works correctly.

It is given a sequence of numbers, separated by a ", ". Iterate through each number by its index, and if the number is smaller or equal to 5, make a multiplication. If the number is larger than 5 and smaller or equal to 10, divide the result by the number. In the end, print the final result.

numbers\_list = int(input()).split(", ")

result = 1

for i in range(numbers\_list):

number = numbers\_list[i+1]

if number <= 5

result \*= number

elif 5 < number <= 10:

result /= number

print(total)

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2, 5, 10 | 1.0 |
| 4, 5, 6, 1, 3 | 10.0 |
| 1, 4, 5 | 20.0 |
| 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 | 0.003968253968253968 |

## Repeat Text

Write a program that receives a **text** on the first line and **times** (to repeat the text) that must be an **integer**. If the user passes a **non-integer** type for the times variable, handle the exception and print a message   
**"Variable times must be an integer"**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Hello  Bye | Variable times must be an integer |
| Hello  2 | HelloHello |

## Value Cannot Be Negative

Create your own exception called ValueCannotBeNegative. Write a program that reads **five numbers** from the console (on separate lines). If a **negative** number occurs, raise the exception.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  4  -5  3  10 | Traceback (most recent call last):  File ".\value\_cannot\_be\_negative.py", line 8, in <module>  raise ValueCannotBeNegative  \_\_main\_\_.ValueCannotBeNegative |