Programming Assignment Uni 3

Irfan Manzoor

University of the People

CS 1101-01: Programming Fundamentals – AY2025-T1

Alok Parihar

27 September 2024

Question 1

## Code:

# Function to count down from a positive number to zero

def countdown(n):

if n <= 0: # Base case: when n reaches zero or negative, print 'Blastoff!'

print('Blastoff!')

else:

print(n) # Print the current number

countdown(n - 1) # Recursively call countdown with n-1

# Function to count up from a negative number to zero

def countup(n):

if n >= 0: # Base case: when n reaches zero or positive, print 'Blastoff!'

print('Blastoff!')

else:

print(n) # Print the current number

countup(n + 1) # Recursively call countup with n+1

# Main function for user interaction

def main():

number = int(input("Enter a number: ")) # Get user input as an integer

if number > 0:

countdown(number) # If input is positive, call countdown

elif number < 0:

countup(number) # If input is negative, call countup

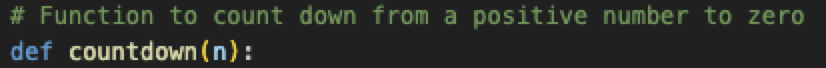
else:

countdown(number) # If input is zero, call countdown (arbitrarily chosen)

if \_\_name\_\_ == "\_\_main\_\_":

main() # Run the main function when the script is executed

**Explanation:**



First, we define a function called `countdown` that takes one argument, `n`. The function is designed to count down from a positive number to zero. It's like counting down the seconds before a rocket launch.



Then, we have an `if` statement that checks if the value of `n` is less than or equal to zero. This is the base case of the countdown function, indicating when the countdown should stop and print "Blastoff!" This is like how a countdown timer reaches zero before something exciting happens, like the start of a race.



If `n` is zero or negative, this line prints "Blastoff!" to signify the end of the countdown. It’s like the moment when a rocket takes off or when a New Year's Eve countdown reaches midnight. :)

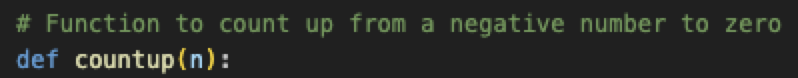
A black background with green text

Description automatically generated

If `n` is still greater than zero, this prints the current value of `n`. Like displaying the number of seconds left in a countdown timer, such as "3, 2, 1..."



The `countdown` function is called recursively with the argument `n-1`. This effectively reduces the value of `n` by 1 and continues the countdown. Like the idea of updating a countdown timer by subtracting one second at a time until it reaches zero.



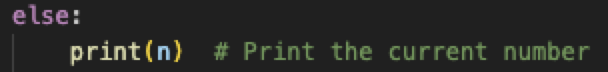
Next, we define a function called `countup`, which takes one argument, `n`. This function is designed to count from a negative number to zero, like counting from negative temperatures to zero.



Like the `countdown` function, this `if` statement checks if `n` has reached zero or become positive, which is the base case for the `countup` function.



When the base case is reached, this line prints "Blastoff!" to signal the end of the countup. This could represent reaching the top floor of a building from the ground level.



If `n` is still negative, this prints the current value of `n`. It's like we see a floor number as we keep going up the building in the elevator display.



The `countup` function is called recursively with the argument `n+1`, incrementing `n` by 1 and continuing the countup process. It's like counting one unit at a time until reaching zero.

A black background with green letters

Description automatically generated

Lastly, we define the `main` function, which serves as the entry point of the program and handles user interactions.



The program prompts the user to enter a number and then reads the input from the keyboard. It converts the input into an integer and stores it in the `number` variable. This is like a user entering a countdown or countup starting point.



This `if` statement checks whether the entered number is greater than zero. If it is, it means the user wants to count down.



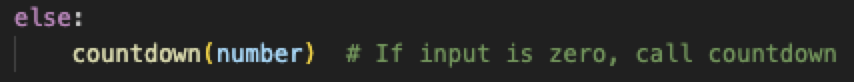
In this case, the `countdown` function is called with the user's input as an argument, initiating the countdown from the entered number.



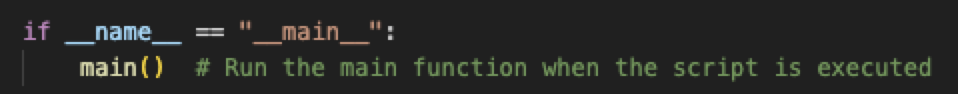
This `elif` statement checks whether the entered number is less than zero. If it is, it means the user wants to count.



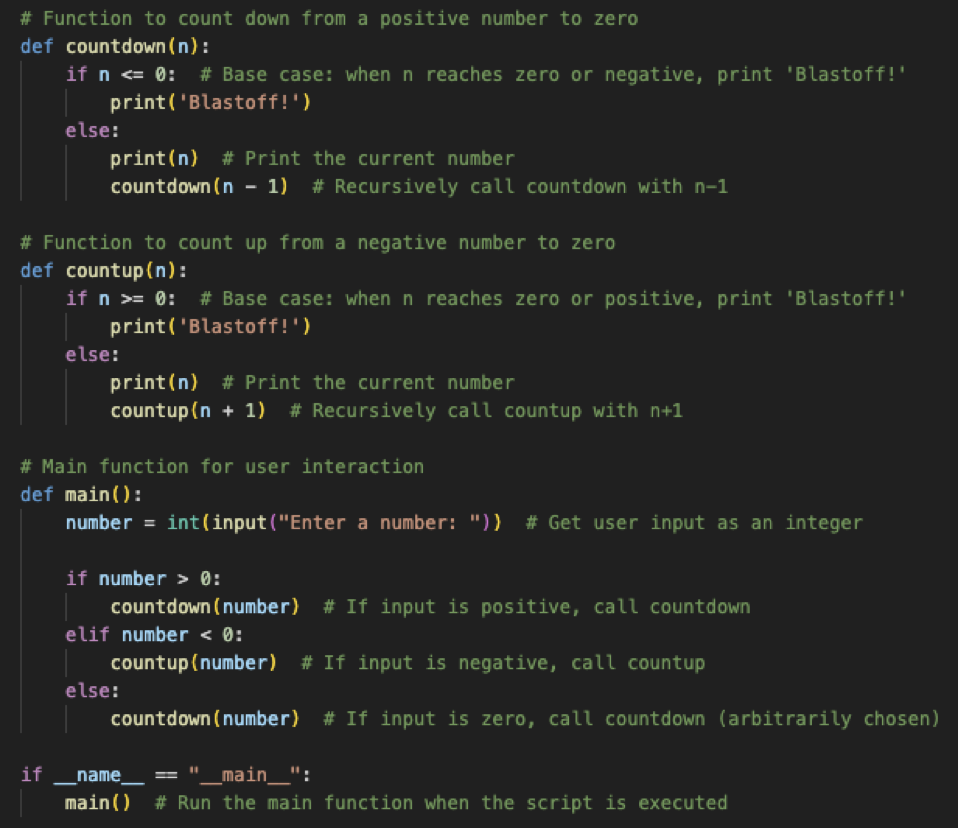
In this case, the `countup` function is called with the user's input as an argument, initiating the countup from the entered negative number.



If the user enters zero, this `else` block is executed, although it could have been set to call `countup` as well. In this case, the `countdown` function is called, effectively treating zero as a countdown.

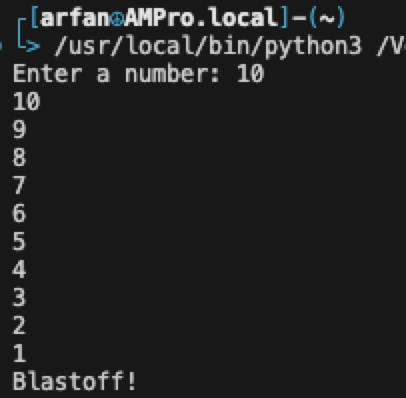


In the end, code ensures that the `main` function is executed when the script is run directly (not when it's imported as a module in another script). It’s like making this code execute as a standalone program. We could also remove this `if` statement and simply run the `main()` function as well.

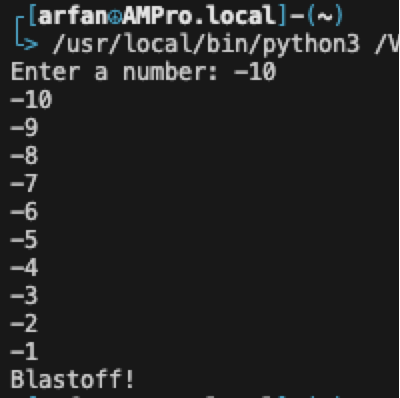


**Outputs:**

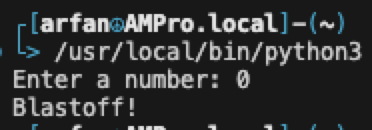
1. Positive Number



1. Negative Number



1. Zero



# Question 2

## Code

def divide\_numbers():

try:

numerator = float(input("Enter the numerator: "))

denominator = float(input("Enter the denominator: "))

result = numerator / denominator

print(f"The result of {numerator} / {denominator} is {result}.")

except ZeroDivisionError as e:

print("Error: Division by zero is not allowed.")

print(f"Exception message: {e}")

except ValueError as e:

print("Error: Please enter valid numbers.")

print(f"Exception message: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

divide\_numbers()

## For Junior Developers

### Understanding the Error:

The code attempts to divide `numerator` by `denominator`. If `denominator` is zero, Python raises a `ZeroDivisionError`.

### Using `try` and `except` Blocks:

* The `try` block allows you to test a block of code for errors.
* The `except` block lets you handle specific errors gracefully. In this case, we've handled `ZeroDivisionError` and `ValueError`.

### Providing Feedback:

When an error occurs, the program outputs an error message, which is crucial for user experience and debugging.

## Fixing the Error

To fix the error, ensure that the denominator is not zero before performing the division. You can modify the program as follows:

def divide\_numbers():

try:

numerator = float(input("Enter the numerator: "))

denominator = float(input("Enter the denominator: "))

if denominator == 0:

raise ValueError("Denominator cannot be zero.")

result = numerator / denominator

print(f"The result of {numerator} / {denominator} is {result}.")

except ZeroDivisionError as e:

print("Error: Division by zero is not allowed.")

print(f"Exception message: {e}")

except ValueError as e:

print("Error: Please enter valid numbers.")

print(f"Exception message: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

divide\_numbers()

**References**

Python Official Documentation. *Python 3.11.5 documentation*. Python.org

<https://docs.python.org/3/>

Python Official Documentation. *Input function (`input()`)*. Python.org

<https://docs.python.org/3/library/functions.html#input>

Python Official Documentation. *Conditional statements (`if`, `elif`, `else`)*.

<https://docs.python.org/3/tutorial/controlflow.html#if-statements>

Real Python. *Recursion in Python*.

<https://realpython.com/python-recursion/>

Programiz. *Pyton Recursion*. Understanding recursion in Python.

<https://www.programiz.com/python-programming/recursion>

Python Official Documentation. *Function Definitions*. Defining functions in Python

<https://docs.python.org/3/tutorial/controlflow.html#defining-functions>

Stack Overflow. *"if \_\_name\_\_ == ‘\_\_main\_\_’:"*. What does if \_\_name\_\_ == "\_\_main\_\_": do?.

<https://stackoverflow.com/questions/419163/what-does-if-name-main-do>

Real Python. *Defining Main Functions in Python*. Understanding Python's main function

<https://realpython.com/python-main-function/>

Real-world analogy for recursion. *Explaining recursion with real-world examples*.

<https://www.cs.utah.edu/~germain/PPS/Topics/recursion.html>

Python Software Foundation. (n.d.). Errors and Exceptions

<https://docs.python.org/3/tutorial/errors.html>

Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media.