**11th June 2023**

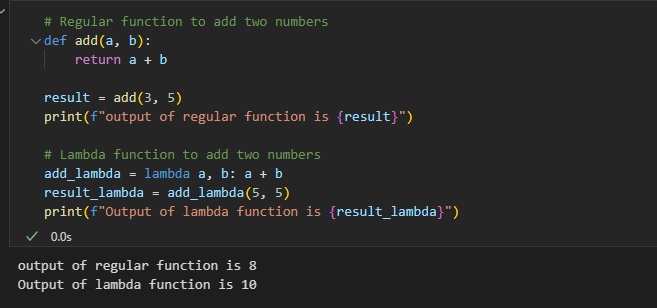
**Assignment - 9**

1. **What is a lambda function in Python, and how does it differ from a regular function?**

**Ans:** In Python, a lambda function is a small anonymous function. It is called "anonymous" because it doesn't have a name like a regular function defined with the def keyword. Instead, lambda functions are defined using the lambda keyword, followed by the function's parameters and a single expression. The result of the expression is automatically returned when the lambda function is called.

Difference between lambda functions and regular functions:

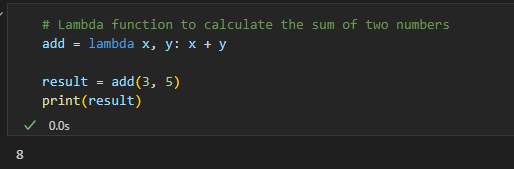
* Syntax: Lambda functions having more compact syntax compared to the regular function defined by ‘def’.
* Function name: Lambda functions do not have any name, while regular functions have a name which can be used to call the function.
* Number of expressions: Lambda functions are limited to the single line functions while regular functions does not having limitations of expressions.
* Example using Lambda function:



1. **Can a lambda function in Python have multiple arguments? If yes, how can you define and use them?**

**Ans:** Yes, a lambda function in Python can have multiple arguments. Lambda functions can take any number of arguments, just like regular functions, but they are defined using a more compact syntax.

**Example:**

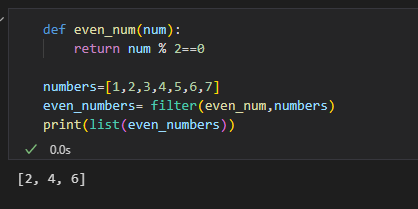
****

1. **How are lambda functions typically used in Python? Provide an example use case.**

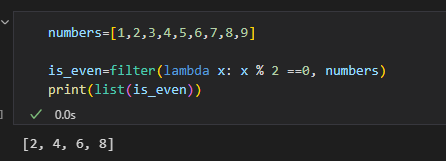
**Ans:** Lambda functions are extremely useful when we require a short function to be used only in limited part of code and don’t want to define a full fledge code using ‘def’ keyword.

Some common use cases for lambda function in Python are:

Suppose we have a list of numbers, and we want to filter out only the even numbers from the list. By using ‘def’ statement we can write code as below.



By using lambda function, we can write above code in one line as below:



1. **What are the advantages and limitations of lambda functions compared to regular functions in Python?**

**Ans: Advantages of Lambda Functions:**

Compact Syntax: Lambda functions allow you to define simple, one-liner functions without the need for a full def block, making the code more compact and easier to read in certain situations.

Anonymous Functions: Lambda functions are anonymous, meaning they don't have a formal name, which can be useful when you only need a function for a short, localized task without cluttering the namespace.

Function Literals: Lambda functions are treated as first-class citizens in Python, which means they can be assigned to variables, passed as arguments to other functions, and returned as values from other functions.

Functional Programming: Lambda functions are often used in functional programming paradigms, where higher-order functions like map(), filter(), and sorted() expect functions as arguments. Lambda functions make it easy to define these functions inline.

Reduced Code Complexity: In certain cases, using lambda functions can lead to less code overhead compared to defining a separate named function.

**Limitations of Lambda Functions:**

Single Expression: Lambda functions can only consist of a single expression. They cannot contain multiple statements or complex logic, limiting their usability for more intricate tasks.

No Statements or Documentation: Lambda functions cannot include statements like print or documentation strings (docstrings). As a result, they are less expressive and less suitable for more complex functions that require multiple statements or detailed documentation.

Limited Readability: While lambda functions can make code more concise, they can also make it harder to read and understand, especially if the lambda expression becomes too complex or if it's used extensively throughout the codebase.

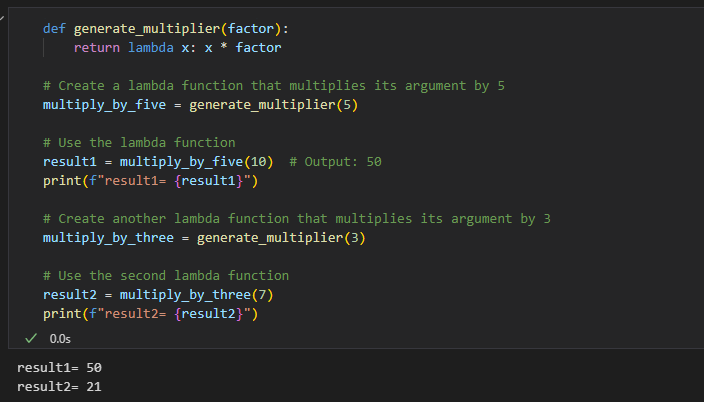
Debugging Challenges: Since lambda functions don't have named identifiers, debugging can be more challenging, as the error traceback will only show the lambda's location in memory, rather than a meaningful function name.

Reduced Reusability: Lambda functions are typically designed for specific, one-time use cases. They lack the reusability and modularity that named functions provide.

1. **Are lambda functions in Python able to access variables defined outside of their own scope? Explain with an example.**

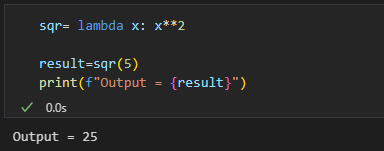
**Ans:** Yes, lambda functions in Python can access variables defined outside of their own scope. Lambda functions have access to the variables from the enclosing lexical scope, just like regular functions.

Here's an example to demonstrate how lambda functions can access variables from their outer scope:



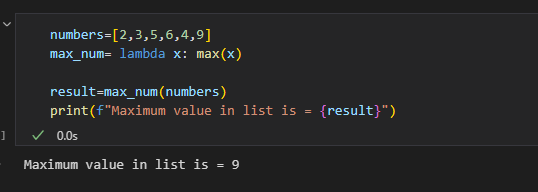
1. **Write a lambda function to calculate the square of a given number.**

**Ans:**

****

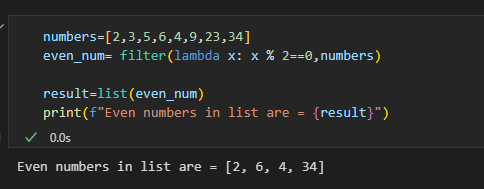
1. **Create a lambda function to find the maximum value in a list of integers.**

**Ans:**

****

1. **Implement a lambda function to filter out all the even numbers from a list of integers.**

**Ans:**

****

1. **Write a lambda function to sort a list of strings in ascending order based on the length of each string.**

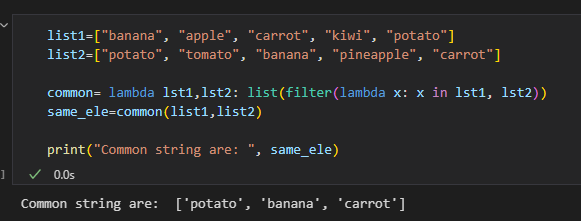
**Ans:**

**A screen shot of a computer program

Description automatically generated**

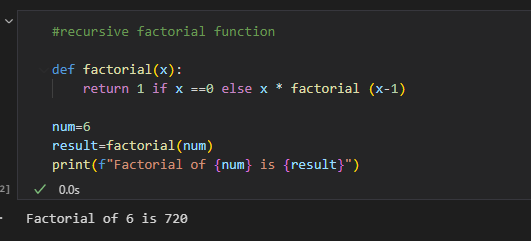
1. **Create a lambda function that takes two lists as input and returns a new list containing the common elements between the two lists.**

**Ans:**

****

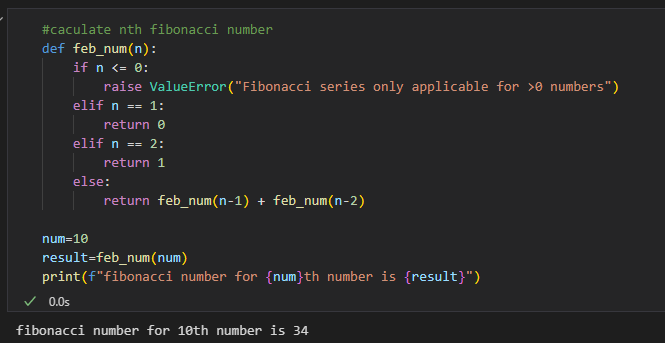
1. **Write a recursive function to calculate the factorial of a given positive integer.**

**Ans:**

****

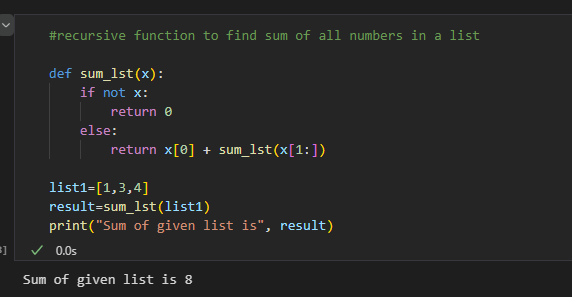
1. **Implement a recursive function to compute the nth Fibonacci number.**

**Ans:**

****

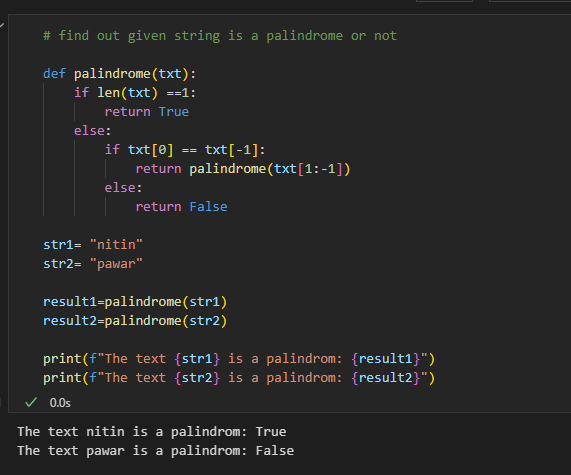
1. **Create a recursive function to find the sum of all the elements in a given list.**

**Ans:**

****

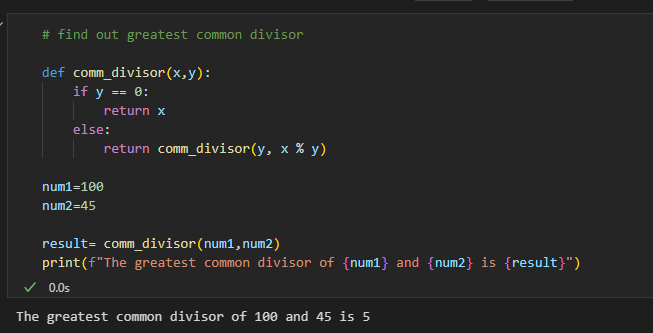
1. **Write a recursive function to determine whether a given string is a palindrome.**

**Ans:**

****

1. **Implement a recursive function to find the greatest common divisor (GCD) of two positive integers**

**Ans:**

****