SR UNIVERSITY

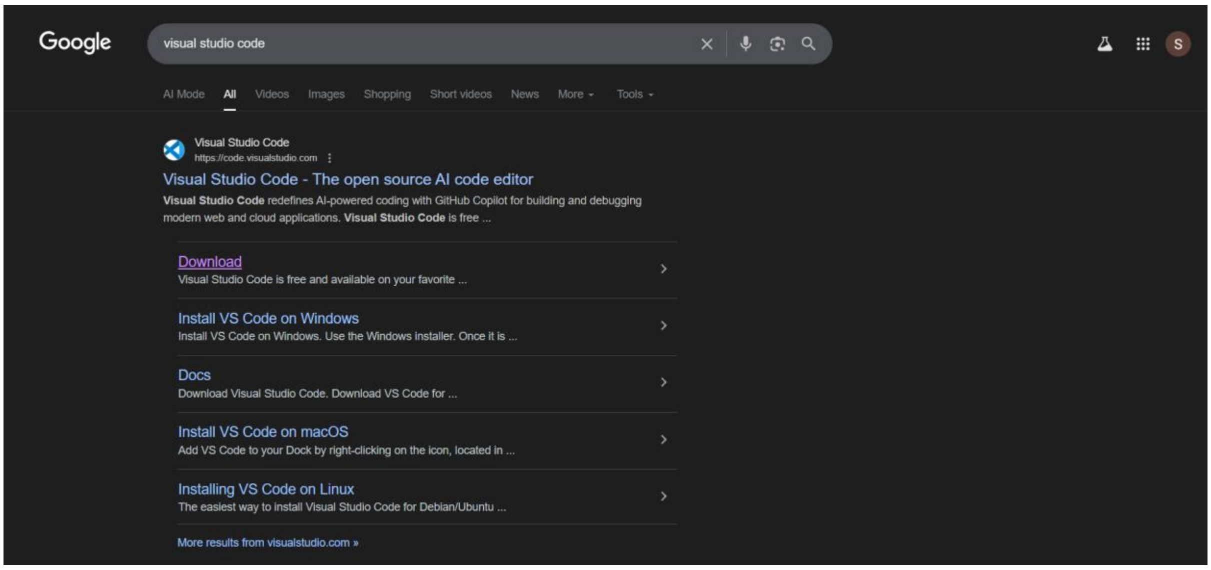
School of Computer Science and Artificial Intelligence

Lab Assignment 1.5

Program	**: B. Tech (CSE)**
Course Title	**: AI Assisted Coding**
Course Code	**: 23CS002PC304**
Semester	**: III**
Academic Session	**: 2025-2026**
Name of Student	**: Akhila**
Enrollment No.	**: 2403A51L40**
Batch No.	**: 52**
Date	**: 09-10-2026**

Task 0

Search the Visual Studio Code in the browser. Then click on the Download.



Visual Studio Code
https://code.visualstudio.com
Visual Studio Code - The open source AI code editor
Visual Studio Code redefines AI-powered coding with GitHub Copilot for building and debugging modern web and cloud applications. **Visual Studio Code** is free ...

Download
Visual Studio Code is free and available on your favorite ...

Install VS Code on Windows
Install VS Code on Windows. Use the Windows installer. Once it is ...

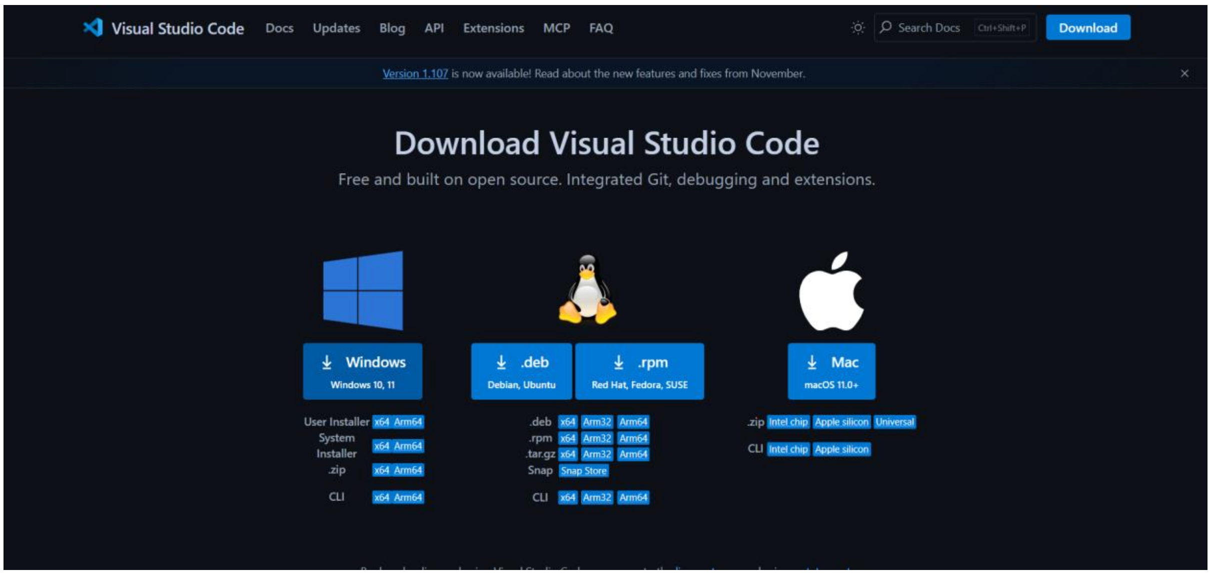
Docs
Download Visual Studio Code. Download VS Code for ...

Install VS Code on macOS
Add VS Code to your Dock by right-clicking on the icon, located in ...

Installing VS Code on Linux
The easiest way to install Visual Studio Code for Debian/Ubuntu ...

More results from visualstudio.com »

Click on the OS your desktop or laptop

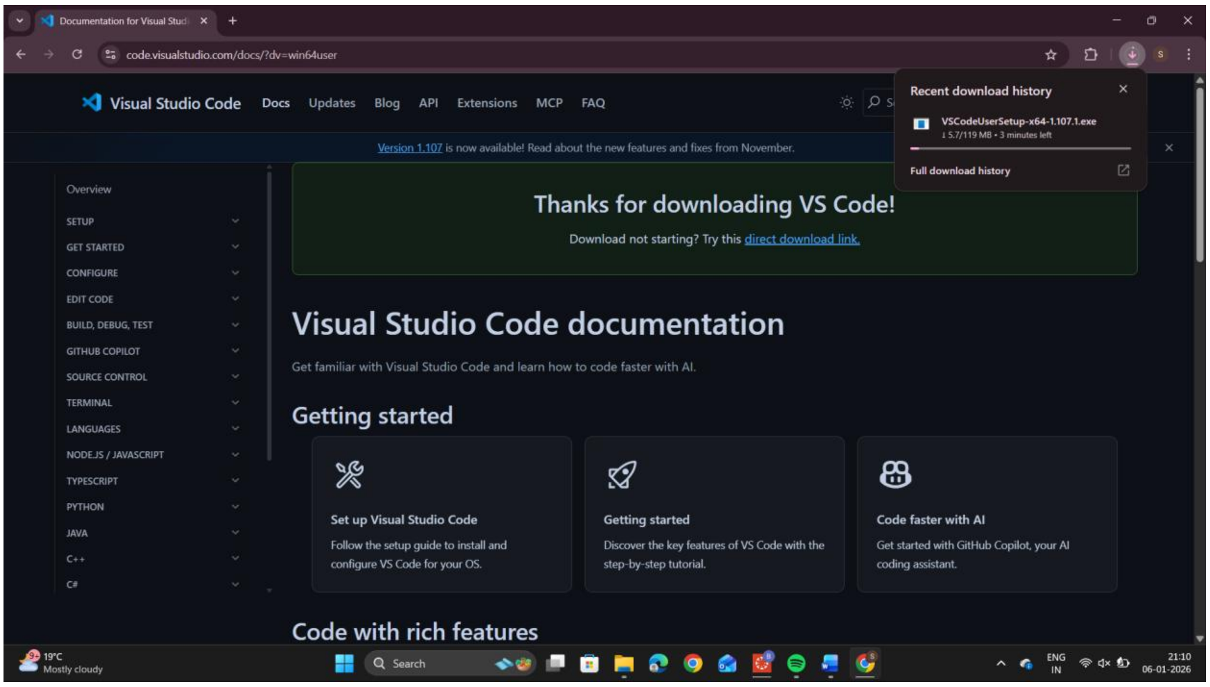


Version 1.107 is now available! Read about the new features and fixes from November.

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

A download .exe file you can see.



Recent download history

VSCodeUserSetup-x64-1.107.1.exe

5.7/119 MB • 3 minutes left

Full download history

code.visualstudio.com/docs/?dv=win64user

Version 1.107 is now available! Read about the new features and fixes from November.

Thanks for downloading VS Code!

Download not starting? Try this direct download link.

Visual Studio Code documentation

Get familiar with Visual Studio Code and learn how to code faster with AI.

Getting started

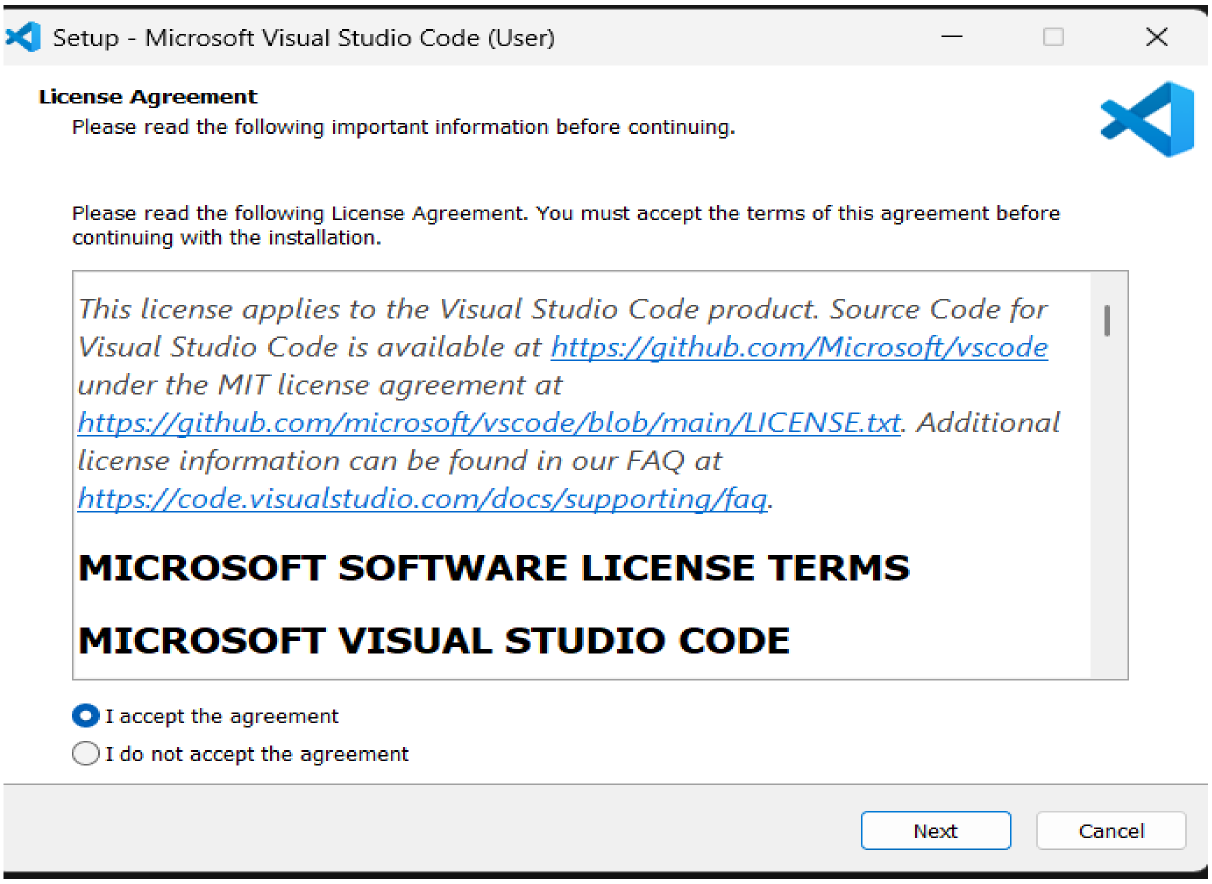
Set up Visual Studio Code
Follow the setup guide to install and configure VS Code for your OS.

Getting started
Discover the key features of VS Code with the step-by-step tutorial.

Code faster with AI
Get started with GitHub Copilot, your AI coding assistant.

Code with rich features

After the downloading the .exe file completed. Click on the .exe file then you will see the below window . Then click on the accept the agreement click next.



Setup - Microsoft Visual Studio Code (User)

License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

This license applies to the Visual Studio Code product. Source Code for Visual Studio Code is available at https://github.com/Microsoft/vscode under the MIT license agreement at https://github.com/microsoft/vscode/blob/main/LICENSE.txt. Additional license information can be found in our FAQ at https://code.visualstudio.com/docs/supporting/faq.

MICROSOFT SOFTWARE LICENSE TERMS

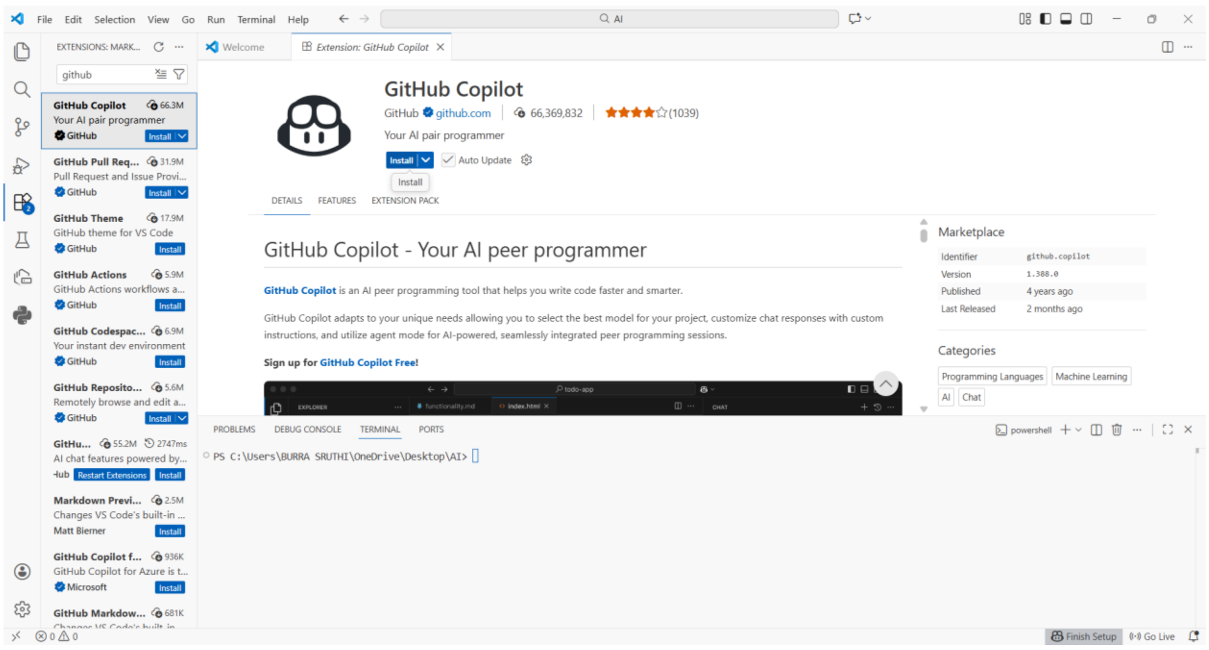
MICROSOFT VISUAL STUDIO CODE

[✓ I accept the agreement]

I do not accept the agreement

After that click next and then click on install. After the completion of download. Open the vs code

Click on the extension then search for GitHub Copilot. Click on the install



GitHub Copilot

GitHub | github.com | 66,369,832 | (1039)

Your AI pair programmer

GitHub Copilot - Your AI peer programmer

GitHub Copilot is an AI peer programming tool that helps you write code faster and smarter.

GitHub Copilot adapts to your unique needs allowing you to select the best model for your project, customize chat responses with custom instructions, and utilize agent mode for AI-powered, seamlessly integrated peer programming sessions.

Sign up for GitHub Copilot Free!

Marketplace

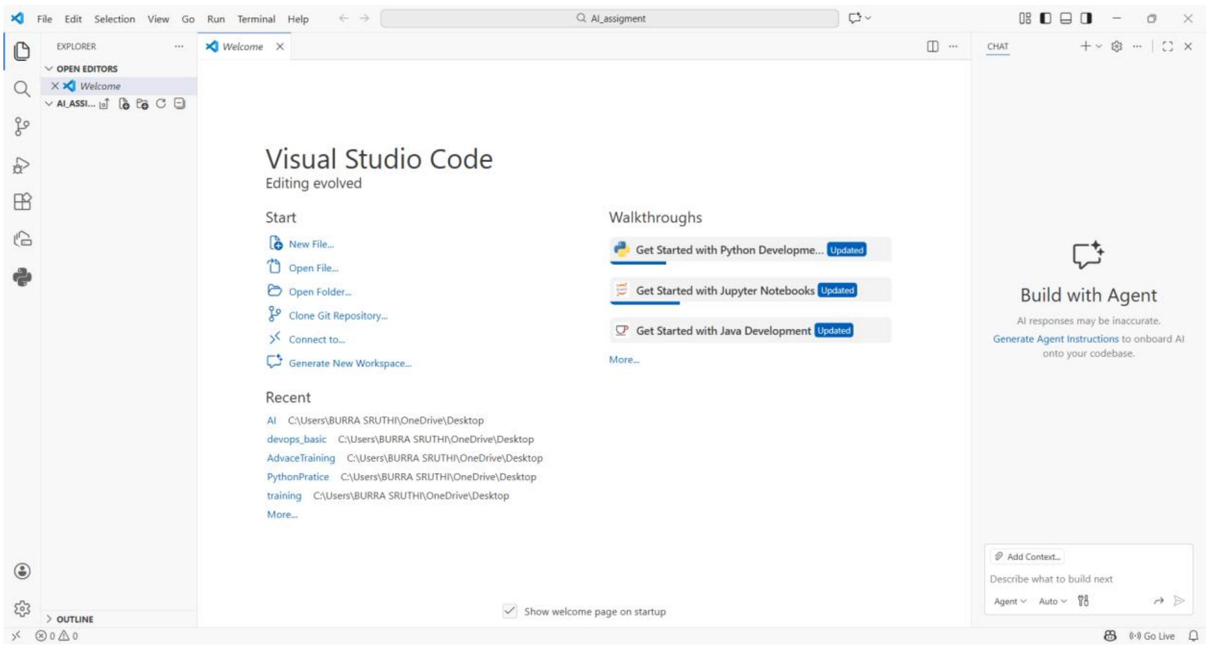
Identifier	github.copilot
Version	1.388.0
Published	4 years ago
Last Released	2 months ago

Categories

Programming Languages, Machine Learning, AI, Chat

```
PS C:\Users\BURRA SRUTHI\OneDrive\Desktop\AI>
```

After the installation completed . sign in into the github copilot using the account of github. Create a new folder on desktop open it in the visual studio.



Visual Studio Code

Editing evolved

Start

New File...
Open File...
Open Folder...
Clone Git Repository...
Connect to...
Generate New Workspace...

Walkthroughs

Get Started with Python Development... Updated
Get Started with Jupyter Notebooks Updated
Get Started with Java Development Updated
More...

Recent

AI	C:\Users\BURRA SRUTHI\OneDrive\Desktop
devops_basic	C:\Users\BURRA SRUTHI\OneDrive\Desktop
AdvaceTraining	C:\Users\BURRA SRUTHI\OneDrive\Desktop
PythonPratice	C:\Users\BURRA SRUTHI\OneDrive\Desktop
training	C:\Users\BURRA SRUTHI\OneDrive\Desktop

More...

Show welcome page on startup

Build with Agent

AI responses may be inaccurate.

Generate Agent Instructions to onboard AI onto your codebase.

Add Context...

Describe what to build next

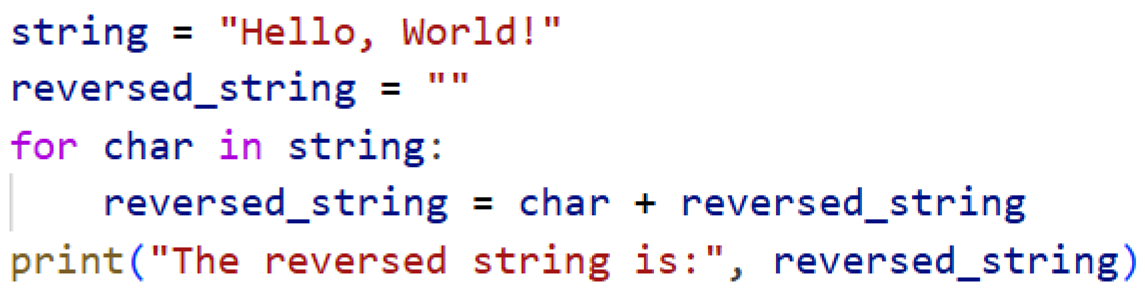
Agent Auto

Create a python program

Task 1

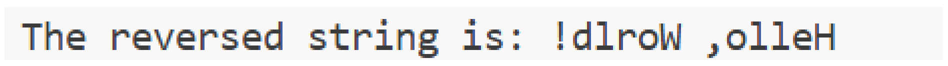
Prompt: Write the python program to reverse a string without using functions

Code:



```
string = "Hello, World!"
reversed_string = ""
for char in string:
    reversed_string = char + reversed_string
print("The reversed string is:", reversed_string)
```

Output:



```
The reversed string is: !dlroW ,olleH
```

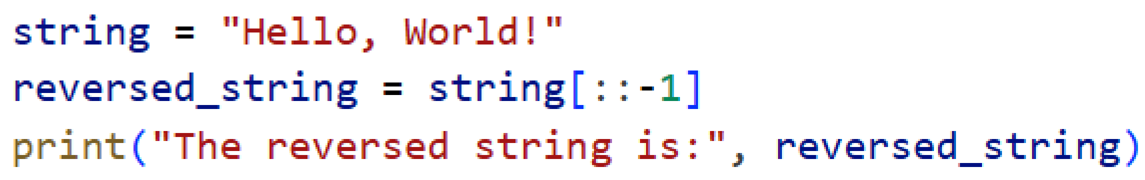
Explanation:

The program reverses a string using a for loop. The original string is stored in a variable, and an empty string is initialized to store the reversed result. Each character of the string is added to the beginning of the new string during iteration, which reverses the order of characters. Finally, the reversed string is printed.

Task 2

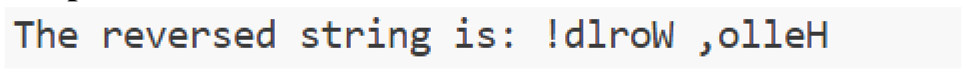
Prompt:Write a python program to reverse a string without using functions and in optimized way

Code:



```
string = "Hello, World!"
reversed_string = string[::-1]
print("The reversed string is:", reversed_string)
```

Output:

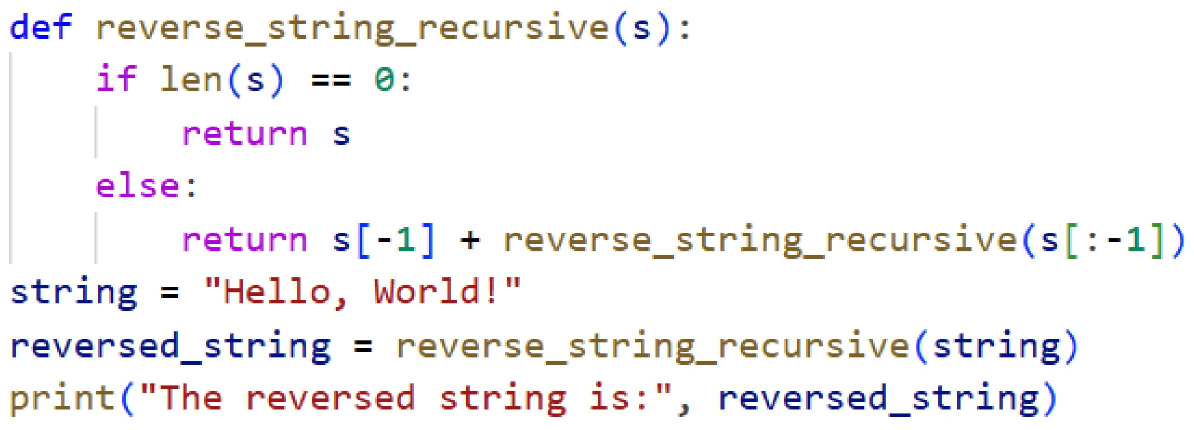


```
The reversed string is: !dlroW ,olleH
```

Explanation:

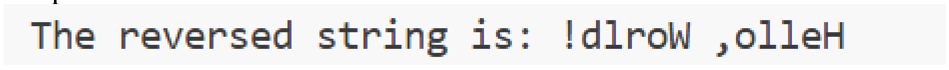
In this program, the string "Hello, World!" is reversed using Python slicing. The slicing syntax [::-1] starts from the end of the string and moves backward, creating a reversed copy of the string. The reversed string is then stored in the variable reversed_string and displayed using the print() function.

Task 3
Prompt:Write a python program to reverse a string using recursion
Code:



```python
def reverse_string_recursive(s):
    if len(s) == 0:
        return s
    else:
        return s[-1] + reverse_string_recursive(s[:-1])
string = "Hello, World!"
reversed_string = reverse_string_recursive(string)
print("The reversed string is:", reversed_string)
```

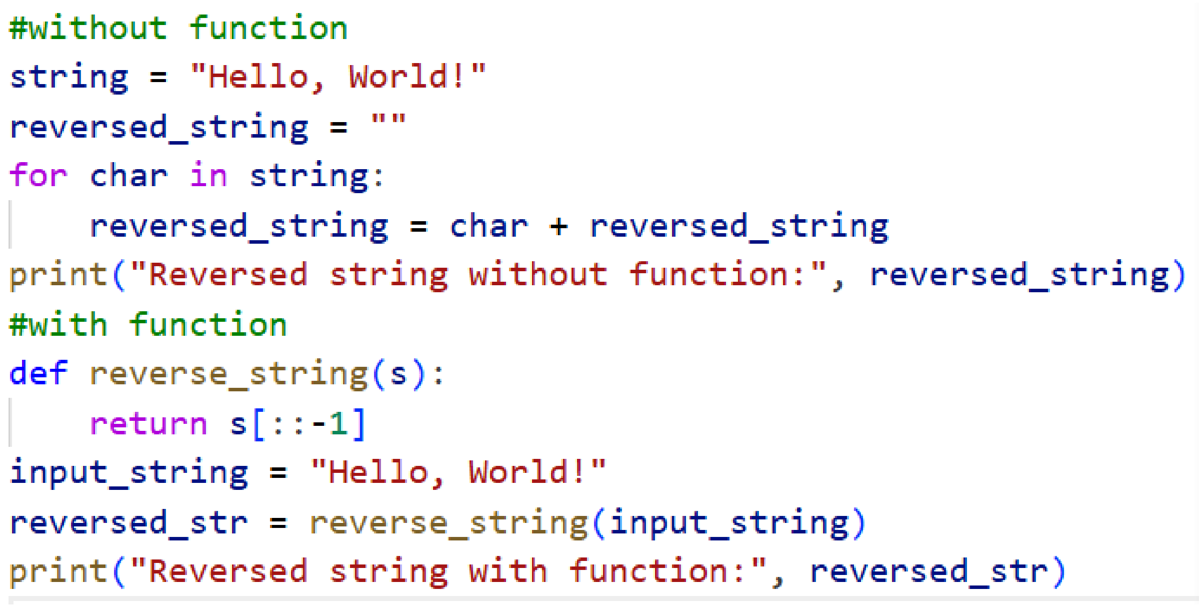
Output:



```
The reversed string is: !dlroW ,olleH
```

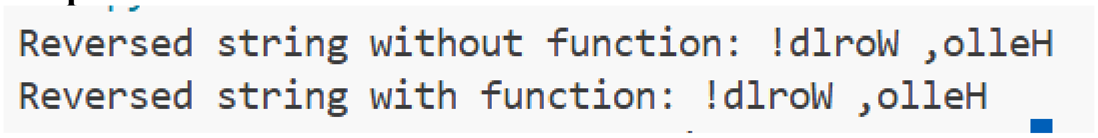
Explanation:
This program reverses a string using recursion. The function reverse_string_recursive() takes a string as input. If the string is empty, it returns the string (base case). Otherwise, the function returns the last character of the string and recursively calls itself with the remaining part of the string excluding the last character. This process continues until the entire string is reversed. Finally, the reversed string is printed.

Task 4
Prompt: Write a python program to reverse a string with and without function
Code:



```python
#without function
string = "Hello, World!"
reversed_string = ""
for char in string:
    reversed_string = char + reversed_string
print("Reversed string without function:", reversed_string)
#with function
def reverse_string(s):
    return s[::-1]
input_string = "Hello, World!"
reversed_str = reverse_string(input_string)
print("Reversed string with function:", reversed_str)
```

Output:



```
Reversed string without function: !dlroW ,olleH
Reversed string with function: !dlroW ,olleH
```

Explanation:

In the first part, the string is reversed **without using a function**. An empty string is initialized, and a for loop is used to traverse each character of the original string. Each character is added at the beginning of the new string, which results in the reversed string.
In the second part, the string is reversed **using a function**. The function uses Python slicing [::-1] to reverse the string in a simple and efficient way. The reversed string is then returned and printed.
Both methods produce the same output, but using a function makes the code more reusable and concise.

Comparative Analysis: With Function vs Without Function

Criteria	**Without Function**	**With Function**
Code Clarity	Logic is explicit and easy to understand step by step.	Code is concise and clean, but slicing may be less intuitive for beginners.
Reusability	Code cannot be reused easily and must be rewritten each time.	Highly reusable; the function can be called multiple times with different inputs.
Debugging Ease	Easier to debug since each step is visible inside the loop.	Slightly harder to debug as the logic is hidden inside slicing.
Suitability for Large-scale Applications	Not suitable, as repeated logic increases code duplication.	More suitable due to modular design and better maintainability.

Task 5

Prompt :Write a Python program to reverse a string using iteration and recursion

Code:

```python
# Using iteration
def reverse_string_iterative(s):
    reversed_string = ""
    for char in s:
        reversed_string = char + reversed_string
    return reversed_string
# Using recursion
def reverse_string_recursive(s):
    if len(s) == 0:
        return s
    else:
        return s[-1] + reverse_string_recursive(s[:-1])
# Test the functions
input_string = "Hello, World!"
reversed_iterative = reverse_string_iterative(input_string)
reversed_recursive = reverse_string_recursive(input_string)
print("Reversed string using iteration:", reversed_iterative)
print("Reversed string using recursion:", reversed_recursive)
```

Output:

```
Reversed string using iteration: !dlroW ,olleH
Reversed string using recursion: !dlroW ,olleH
```

Explanation:

This program demonstrates two different methods to reverse a string: **iteration** and **recursion**.

1. Iterative Approach

The function reverse_string_iterative() takes a string as input and initializes an empty string to store the reversed result. A for loop iterates through each character of the input string. During each iteration, the current character is added to the beginning of the new string, which gradually forms the reversed string. The final reversed string is then returned.

2. Recursive Approach

The function reverse_string_recursive() reverses the string using recursion. If the length of the string is zero, the function returns the string (base case). Otherwise, it takes the last character of the string and calls itself with the remaining substring excluding the last character. This process continues until the base case is reached, resulting in the reversed string.

3. Function Testing

The input string "Hello, World!" is passed to both functions. The returned reversed strings from the iterative and recursive methods are stored in separate variables and printed. Both methods produce the same output.

Comparative Analysis

Criteria	Iterative Approach	Recursive Approach
Execution Flow	Executes sequentially using a loop.	Executes through repeated function calls until the base case is reached.
Time Complexity	O(n), where n is the length of the string.	O(n), but with additional overhead due to recursive calls.
Performance for Large Inputs	More efficient and faster, as it avoids function call overhead.	Less efficient and may cause stack overflow for very large strings.
Memory Usage	Uses constant extra memory apart from the result string.	Uses extra stack memory for each recursive call.
Appropriate Usage	Suitable for practical and large-scale applications.	Suitable for learning recursion and problem decomposition.