

# PYTHON PROGRAMMING

## Assignment # 01

Submitted By: Maria Farooq

### Question 01

#### Part 01

**Write a program to get a string from user and then reverse the string.**

#### Method 01:

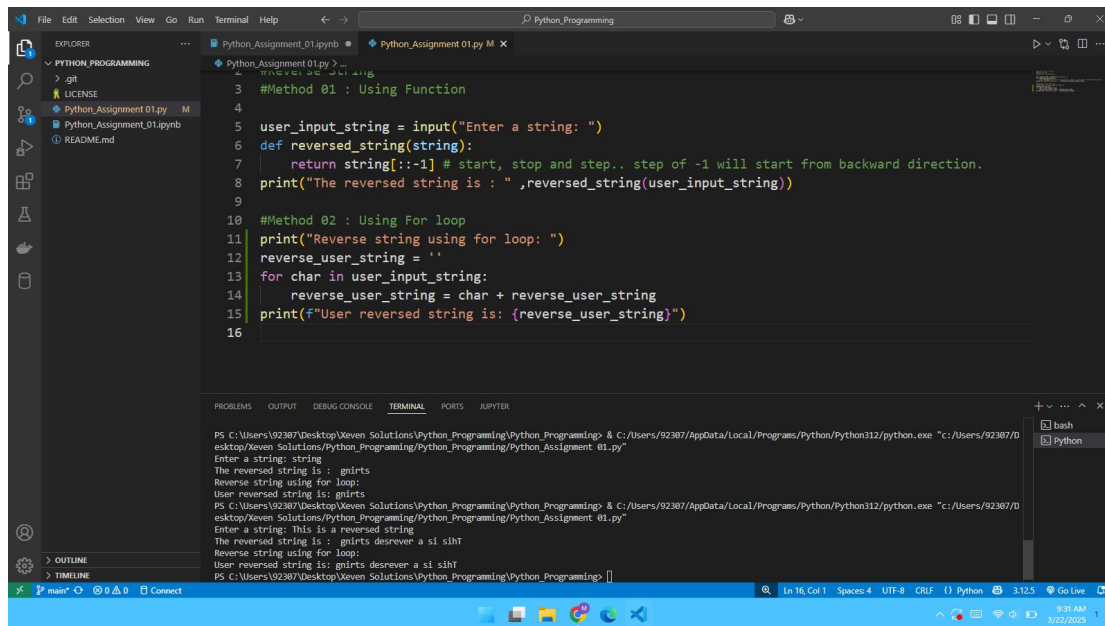
I have used two different methods in it. The first one is by defining a function to get the string from user and then reverse that string. This function use slicing in which 3 values start, stop, step are used. As we want to reverse the string so we will move backward , for this -1 will be used as a step that tells about the direction to start from the end.

```
user_input_string = input("Enter a string: ")
def reversed_string(string):
    return string[::-1] # start, stop and step.. step
of -1 will start from backward direction.
print("The reversed string is :
", reversed_string(user_input_string))

#Method 02 : Using For loop
print("Reverse string using for loop: ")
reverse_user_string = ''
for char in user_input_string:
    reverse_user_string = char + reverse_user_string
print(f"User reversed string is:
{reverse_user_string}")
```

#### Method 02:

The second method is by using for loop in which I have created an empty variable and then store the each character of the string in that variable.



```
Python_Assignment_01.py
1 #Question 01 : Part 02
2 #Numbers of vowels in the string
3 #Method 01 : Using Function
4
5 user_input_string = input("Enter a string: ")
6 def reversed_string(string):
7     return string[::-1] # start, stop and step.. step of -1 will start from backward direction.
8 print("The reversed string is : ",reversed_string(user_input_string))
9
10 #Method 02 : Using For loop
11 print("Reverse string using for loop: ")
12 reverse_user_string = ''
13 for char in user_input_string:
14     reverse_user_string = char + reverse_user_string
15 print(f"User reversed string is : {reverse_user_string}")
16
```

Terminal Output:

```
PS C:\Users\92387\Desktop\Xeven Solutions\Python_Programming\Python_Programming> & C:\Users\92387\AppData\Local\Programs\Python\Python312\python.exe "c:\Users\92387\Desktop\Xeven Solutions\Python_Programming\Python_Programming\Python_Assignment_01.py"
Enter a string: string
The reversed string is : gnirts
Reverse string using for loop:
User reversed string is: gnirts
PS C:\Users\92387\Desktop\Xeven Solutions\Python_Programming\Python_Programming> & C:\Users\92387\AppData\Local\Programs\Python\Python312\python.exe "c:\Users\92387\Desktop\Xeven Solutions\Python_Programming\Python_Programming\Python_Assignment_01.py"
Enter a string: This is a reversed string
The reversed string is : gnirts dsrever a si siht
Reverse string using for loop:
User reversed string is: gnirts dsrever a si siht
PS C:\Users\92387\Desktop\Xeven Solutions\Python_Programming\Python_Programming>
```

## Part 02

### Count the number of vowels in the string.

#### Method 01:

I have used two different methods for this part as well. The first one is using a for loop in which each character of string is stored in a variable named “vowel” and then a function named “lower()” is used to make the characters in lowercase to check that either these characters are present in list `['a','e','i','o','u']` or not. If the character is in vowel list then the character will be added to the variable named “sum\_of\_vowels” and will be displayed.

```
#Question 01 : Part 02
#Numbers of vowels in the string
user_input_string = input("Enter a string: ")
#Method 01
sum_of_vowels = ''
for vowel in user_input_string:
    if vowel.lower() in ['a','e','i','o','u']:
        print("Number of vowels: " , vowel)
        sum_of_vowels += vowel
print("Number of vowels: " , len(sum_of_vowels))
#Method 02 : Using a Function
print("Method 02 : Counting number of vowels using a function:")
```

```
def check_vowels(string):
    vowels = "aeiouAEIOU"
    return sum(1 for char in string if char in vowels)
print("The vowels in the string: " ,
check_vowels(user_input_string))
```

## Method 02:

The second method is using a function that take a “string” as a parameter and check that the string character is in vowel variable and will add 1 for each character.

```
21 #Question 01 : Part 02
22 #Numbers of vowels in the string
23 user_input_string = input("Enter a string: ")
24 #Method 01
25 sum_of_vowels = ''
26 for vowel in user_input_string:
27     if vowel.lower() in ['a','e','i','o','u']:
28         print("Number of vowels: " , vowel)
29         sum_of_vowels += vowel
30 print("Number of vowels: " , len(sum_of_vowels))
31 #Method 02 : Using a Function
32 print("Method 02 : Counting number of vowels using a function:")
33
34 def check_vowels(string):
35     vowels = "aeiouAEIOU"
36     return sum(1 for char in string if char in vowels)
37 print("The vowels in the string: " , check_vowels(user_input_string))
```

Terminal Output:

```
PS C:\Users\92387\Desktop\Xeven Solutions\Python_Programming\Python_Programming> & C:/Users/92387/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/92387/Desktop/Xeven Solutions/Python_Programming/Python_Programming/Python_Assignment_01.py"
Enter a string: This is a string
Number of vowels: i
Number of vowels: a
Number of vowels: i
Number of vowels: a
Method 02 : Counting number of vowels using a function:
The vowels in the string: 4
PS C:\Users\92387\Desktop\Xeven Solutions\Python_Programming\Python_Programming>
```

## Question 02

Problem: Create a Python program that:

- Takes an input number from the user.
- Checks whether the number is even or odd.
- Prints the result.

```
# Question 02

number = int(input("Enter a number: "))
def check_number(number):
    if number % 2 == 0:
        print(f"{number} is an even number")

    else:

        print(f"{number} is an odd number.")
check_number(number) #When we use print statement
inside the function, we do not need to use print here
again.
```

The screenshot shows a Visual Studio Code editor window with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'Python\_Programming' with files like 'Conditional\_statements.ipynb', 'Python\_Assignment\_01.ipynb', and 'Assignment\_01\_Q\_2.py'. The main editor window displays the Python code from the previous block. The terminal at the bottom shows the command prompt running the script, which prompts for a number and outputs '3 is an odd number'.

```
1 # Question 02 : Part 01
2
3 number = int(input("Enter a number: "))
4
5 def check_number(number):
6     if number % 2 == 0:
7         print(f"{number} is an even number")
8
9     else:
10
11         print(f"{number} is an odd number.")
12
13
14 check_number(number) #When we use print statement inside the function, we do not need to use print h
```

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS C:\Users\92307\Desktop\Xeven Solutions\Python\_Programming> & C:\Users\92307\AppData\Local\Programs\Python\Python312\python.exe "c:\Users\92307\Desktop\Xeven Solutions\Python\_Programming\Python\_Programming\Assignment\_01\_Q\_2.py"

Enter a number: 3

3 is an odd number.

PS C:\Users\92307\Desktop\Xeven Solutions\Python\_Programming>

### Question 3:

#### Virtual Environment Application Problem:

Create a Python program that:

1. Takes a list of integers as input.
2. Creates a new virtual environment called sortenv.
3. Installs a package (such as numpy) in the virtual environment.
4. Sorts the list using a numpy method (numpy.sort()).
5. Prints the sorted list.

- Takes a list of integers as input.

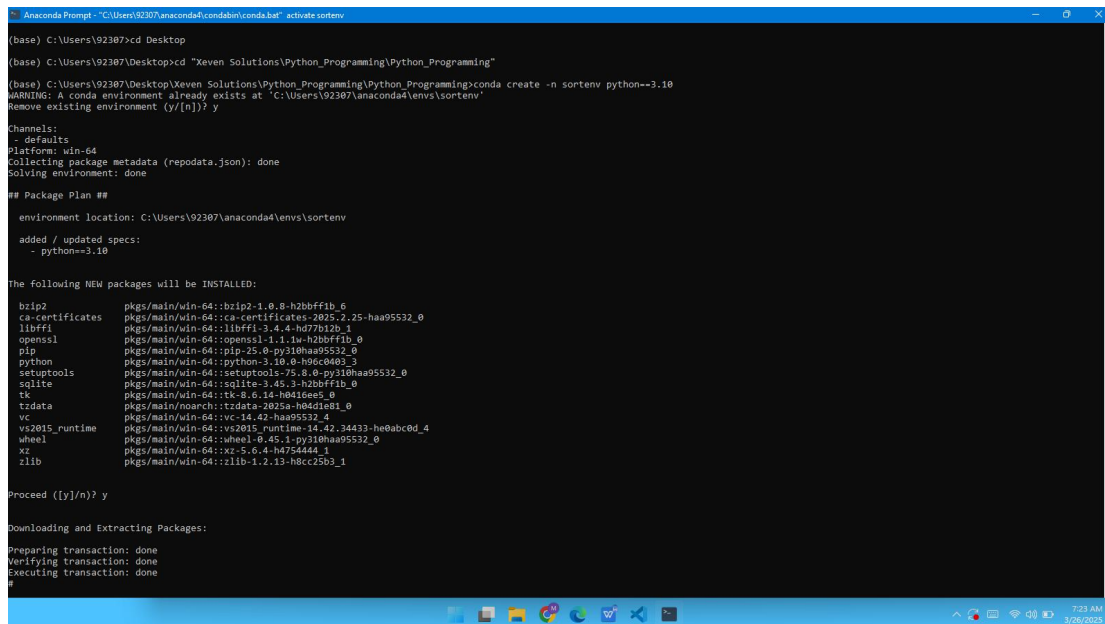
- 2. Creates a new virtual environment called sortenv.

```
conda create -n sortenv python==3.10
conda activate sortenv
```

- 3. Installs a package (such as numpy) in the virtual environment.

```
pip install numpy==2.2
```

- 4. Sorts the list using a numpy method (numpy.sort()).
- 5. Prints the sorted list.



```
Anaconda Prompt - "C:\Users\92307\anaconda\conda\bin\conda.bat" activate sortenv
(base) C:\Users\92307>cd Desktop
(base) C:\Users\92307\Desktop>cd "Xeven Solutions\Python_Programming\Python_Programming"
(base) C:\Users\92307\Desktop\Xeven Solutions\Python_Programming\Python_Programming>conda create -n sortenv python==3.10
WARNING: A conda environment already exists at 'C:\Users\92307\anaconda4\envs\sortenv'
Remove existing environment (y/[n])? y
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\92307\anaconda4\envs\sortenv
added / updated specs:
  - python==3.10

The following NEW packages will be INSTALLED:

bzip2                pkgs/main/win-64::bzip2-1.0.8-h2bbff1b_6
ca-certificates      pkgs/main/win-64::ca-certificates-2025.2.25-haa95532_0
libffi               pkgs/main/win-64::libffi-3.4.4-hd77b12b_1
openssl              pkgs/main/win-64::openssl-1.1.1w-h2bbff1b_0
pip                  pkgs/main/win-64::pip-25.0-py310haa95532_0
python               pkgs/main/win-64::python-3.10.0-h8c04083_3
setuptools           pkgs/main/win-64::setuptools-76.8.0-py310haa95532_0
sqlite               pkgs/main/win-64::sqlite-3.45.3-h2bbff1b_0
tk                   pkgs/main/win-64::tk-8.6.14-h8416ees_0
tzdata               pkgs/main/noarch::tzdata-2025a-hb851e81_0
vc                   pkgs/main/win-64::vc-14.42-haa95532_4
vs2015_runtime       pkgs/main/win-64::vs2015_runtime-14.42.34433-he0abc0d_4
wheel                pkgs/main/win-64::wheel-0.45.1-py310haa95532_0
xz                   pkgs/main/win-64::xz-5.6.4-h4754444_1
zlib                 pkgs/main/win-64::zlib-1.2.13-h8cc25b3_1

Proceed ([y]/n)? y

Downloading and Extracting Packages:
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

```
Anaconda Prompt - "C:\Users\92307\Anaconda3\condabin\conda.bat" activate sortenv
sqlite pkgs/main/win-64::sqlite-3.45.3-h2bfff1b_0
tk pkgs/main/win-64::tk-8.6.14-h841eees_0
tzdata pkgs/main/noarch::tzdata-2025a-h86d1e81_0
vc pkgs/main/win-64::vc-14.42-haa95532_4
vs2015_runtime pkgs/main/win-64::vs2015_runtime-14.42.34433-h90abc8d_4
wheel pkgs/main/win-64::wheel-0.45.1-py310haa95532_0
xz pkgs/main/win-64::xz-5.6.4-h4754444_1
zlib pkgs/main/win-64::zlib-1.2.13-h8cc25b3_1

Proceed ([y]/n)? y

Downloading and Extracting Packages:
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate sortenv
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) C:\Users\92307\Desktop\Xeven Solutions\Python_Programming\Python_Programming>conda activate sortenv
(sortenv) C:\Users\92307\Desktop\Xeven Solutions\Python_Programming\Python_Programming>pip install numpy==2.2
Collecting numpy==2.2
  Using cached numpy-2.2.0-cp310-cp310-win_amd64.whl.metadata (60 kB)
Using cached numpy-2.2.0-cp310-cp310-win_amd64.whl (12.9 MB)
Installing collected packages: numpy
Successfully installed numpy-2.2.0

(sortenv) C:\Users\92307\Desktop\Xeven Solutions\Python_Programming\Python_Programming>
```

Using Anaconda:

```
#Using Anaconda
import numpy as np

#Virtual environment setup
def setup_virtual_environment():
    print("Create virtual environment using anaconda
by running this command: ")
    print("conda create -n sortenv python==3.12 ")
    print("Activate Environment: ")
    print("conda activate sortenv")
    print("Install Python library: Numpy")
    print("!pip install numpy")
```

```
#Sorting array
def sort_list(arr):
    return np.sort(arr)
```

```
#Taking input from user
try:
    user_input = input("Enter a list of integers
seperated by space: ")
```

```

    num_list = list(map(int, user_input.split()))
    print(setup_virtual_environment())
    print("Sorted list: " , sort_list(num_list))
except ValueError:
    print("Invalid Input!")

```

Method 02:

```

def sort_list(arr):
    return np.sort(arr)
# Simulating virtual environment setup
def setup_virtual_env():
    print('Step 1: Create a virtual environment
using:')
    print(" Python -m venv sortenv")
    print('Step 2: Activate the virtual envorinment')
    print(' Windows: sortenv\\Scripts\\activate')
    print(' macOS/Linux: source
sortenv/bin/activate')
    print('Step 3: install NumPy in the virtual
environment')
    print(' pip install numpy')
    print('Step 4: Now, sorting the list using
NumPy...')

Taking user input as a list of integers
try:
    user_input = input('Enter a list of numbers
separated by spaces: ')
    num_list = list(map(int,user_input.split()))

    setup_virtual_env()

    print('Sorted list: ',sort_list(num_list))

```

```
except ValueError:  
    print('Invalid input! Please enter a list of  
integers.')
```