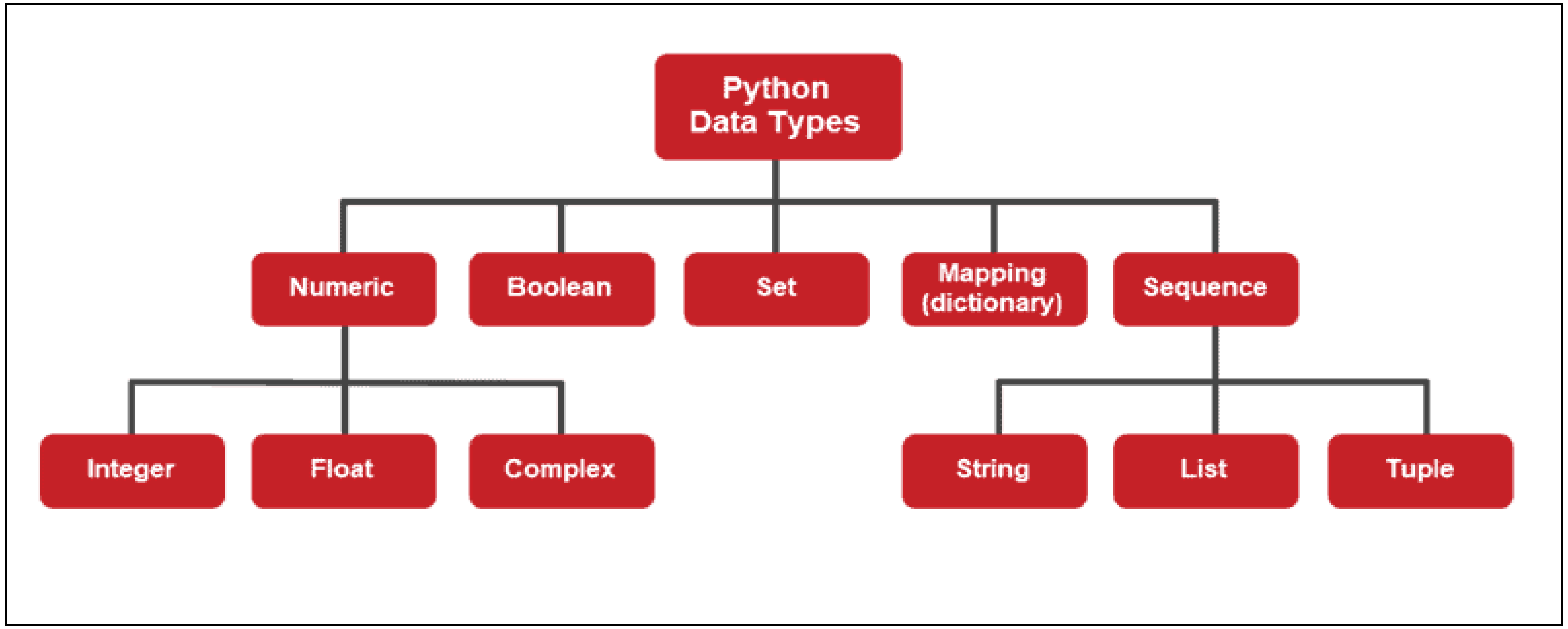
STRINGS in Python

Introduction

- Previously, we have talked about sequence data types, which is an orderly collection of items and each item is indexed by an integer



Strings

- String is a sequence which is made up of one or more UNICODE characters. Here the character can be a letter, digit, whitespace or any other symbol.
- A string can be created by enclosing one or more characters in single, double or triple quote.

  e.g.)

  - ‘Hello World’

  - “gdesv93@&u”

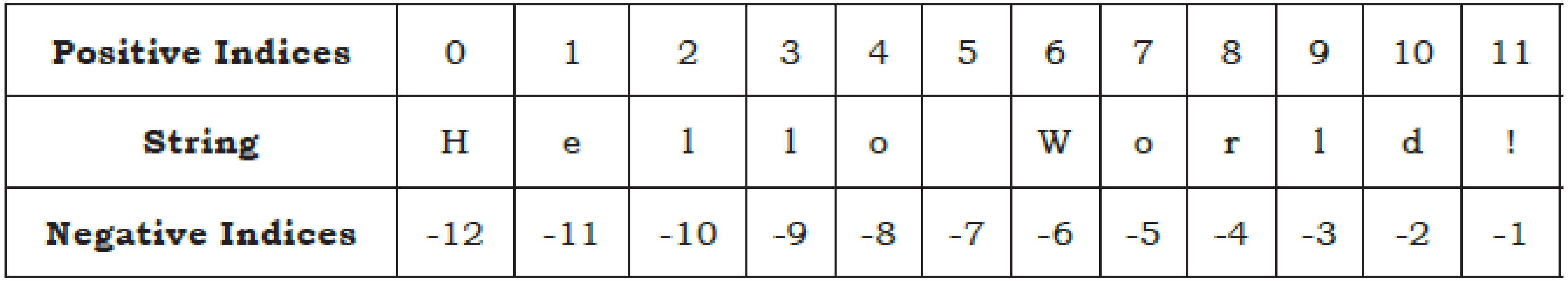
  - ‘‘‘Hello, how are you?’’’

  - “““ I am fine.”””

Accessing characters in a String

- Each individual character in a string can be accessed using a technique called indexing.
- The index of the first character (from left) in the string is 0 and the last character is n-1 where n is the length of the string.

Table 8.1 Indexing of characters in string 'Hello World!'



Positive Indices	0	1	2	3	4	5	6	7	8	9	10	11
String	H	e	l	l	o		W	o	r	l	d	!
Negative Indices	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Accessing characters in a String

- The index specifies the character to be accessed in the string and is written in square brackets ([ ]).

  e.g.)

```
str1 = "Hello World"

str1[3]
'l'

str1[0]
'H'
```

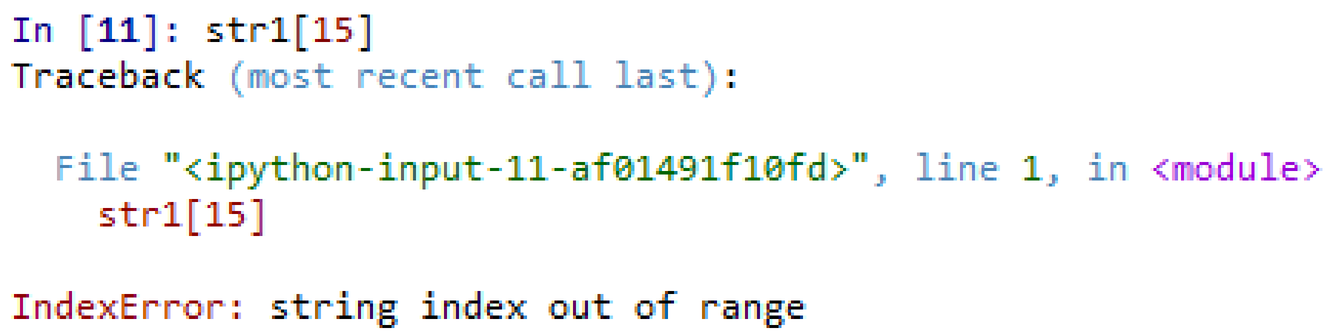
- The index can also be an expression including variables and operators but the expression must evaluate to an integer. For e.g.)

```
str1[2+1]
'l'
```

Accessing characters in a String

- If we give index value out of this range then we get an *IndexError*.

  For e.g.)

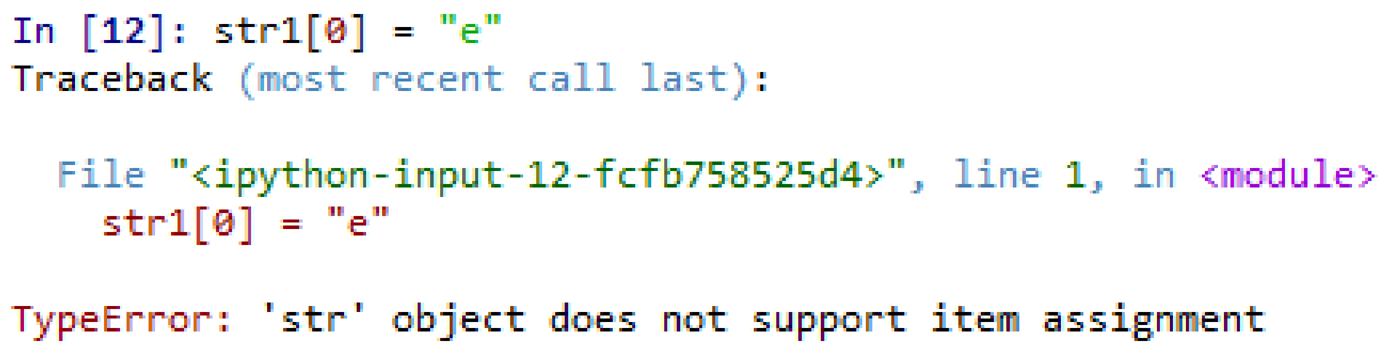


```
In [11]: str1[15]
Traceback (most recent call last):

  File "<ipython-input-11-af01491f10fd>", line 1, in <module>
    str1[15]

IndexError: string index out of range
```

String is immutable

- A string is an immutable data type. It means that the contents of the string cannot be changed after it has been created.
- If we try to change the content of a string, it would lead to an error.

For e.g.)



```
In [12]: str1[0] = "e"
Traceback (most recent call last):

  File "<ipython-input-12-fcfb758525d4>", line 1, in <module>
    str1[0] = "e"

TypeError: 'str' object does not support item assignment
```

Operations on string: Concatenation

- To concatenate means to join. Python allows us to join two strings using concatenation operator plus which is denoted by symbol +.

For e.g.)

```
str1 = "Hello"

str2 = "World"

str1 + str2
'HelloWorld'
```

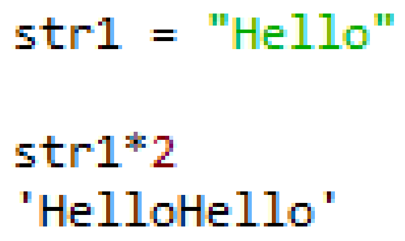
Note:

After the concatenation operation, there will be no change in the values of str1 and str2

Operations on string: Repetition

- Python allows us to repeat the given string using repetition operator which is denoted by symbol *.

For e.g.)



```
str1 = "Hello"

str1*2
'HelloHello'
```

Note:

After the repetition operation, there will be no change in the values of str1

Operations on string: Membership

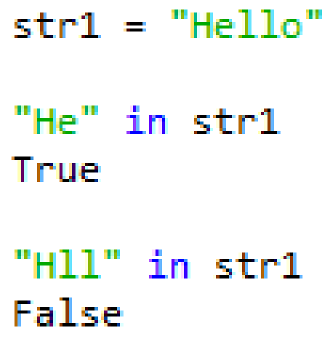
- Python has two membership operators:

  - 'in' and

  - 'not in'

- The 'in' operator takes two strings and returns True if the first string appears as a substring in the second string, otherwise it returns False.

  For e.g.)

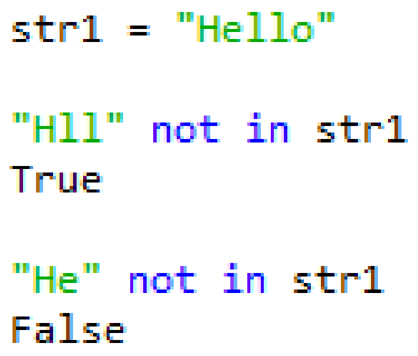


```
str1 = "Hello"

"He" in str1
True

"Hll" in str1
False
```

Operations on string: Membership

- The 'not in' operator also takes two strings and returns True if the first string does not appear as a substring in the second string, otherwise returns False.

  For e.g.)

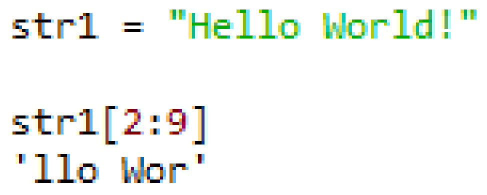


```
str1 = "Hello"

"Hll" not in str1
True

"He" not in str1
False
```

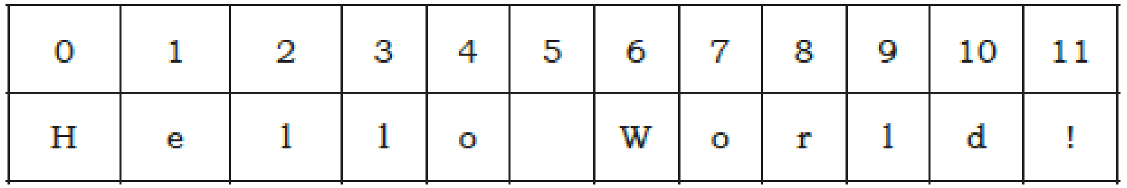
Operations on string: Slicing

- In Python, to access some part of a string or substring, we use a method called slicing. This can be done by specifying an index range.
- Given a string str1, the slice operation str1[n:m] returns the part of the string str1 starting from index n (inclusive) and ending at m (exclusive).

  For e.g.)



```
str1 = "Hello World!"

str1[2:9]
'llo Wor'
```



0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o		W	o	r	l	d	!

- In other words, we can say that str1[n:m] returns all the characters starting from str1[n] till str1[m-1].

Operations on string: Slicing

Note:

The numbers of characters in the substring will always be equal to difference of two indices m and n, i.e., (m-n).

- If the first index is not mentioned, the slice starts from index. For e.g.)

```
str1[:5]
'Hello'
```

- If the second index is not mentioned, the slicing is done till the length of the string.

For e.g.)

```
str1[2:]
'llo World!'
```

Operations on string: Slicing

- The slice operation can also take a third index that specifies the 'step size'. For example, str1[n:m:k], means every kth character has to be extracted from the string str1 starting from n and ending at m-1.

For e.g.)

```
str1[1:10:2]
'el ol'
```

- By default, the step size is one and negative indexes can also be used for slicing.

For e.g.)

```
str1[-6:-1]
'World'
```

H	e	l	l	o		W	o	r	l	d	!
-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

String methods and built-in functions

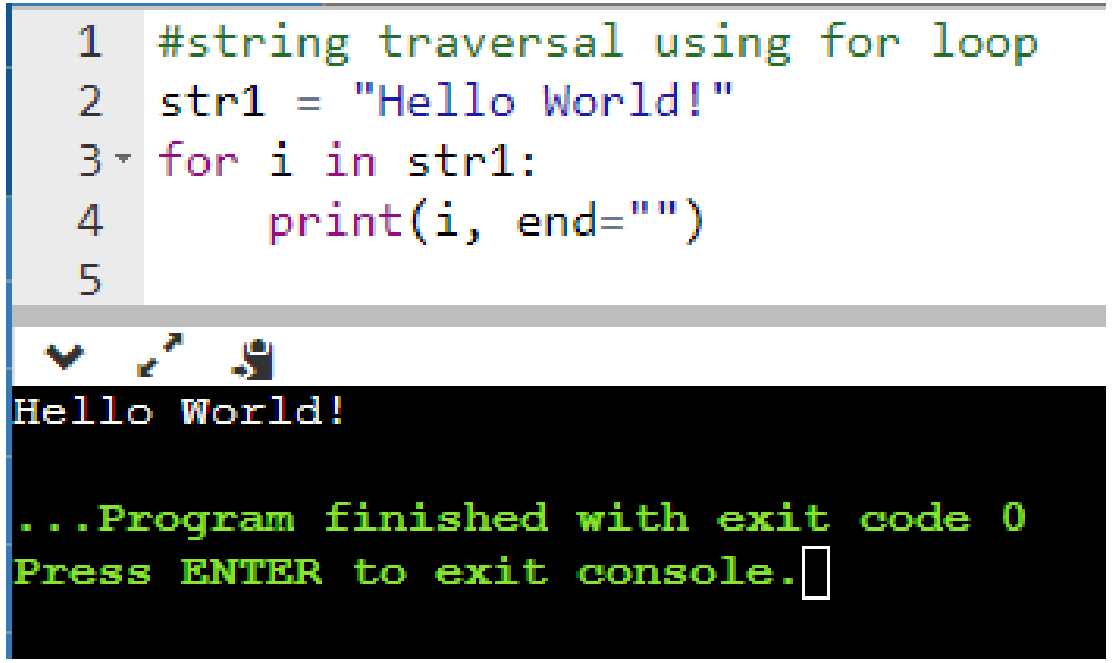
- **built-in functions:**

  len(), capitalize(), title(), lower(), upper(), count(), find(), index(), endswith(), startswith(), isalnum(), isalpha(), isdigit(), islower(), isupper(), isspace(), lstrip(), rstrip(), strip(), replace(), join(), partition(), split()

Method	Description	Example

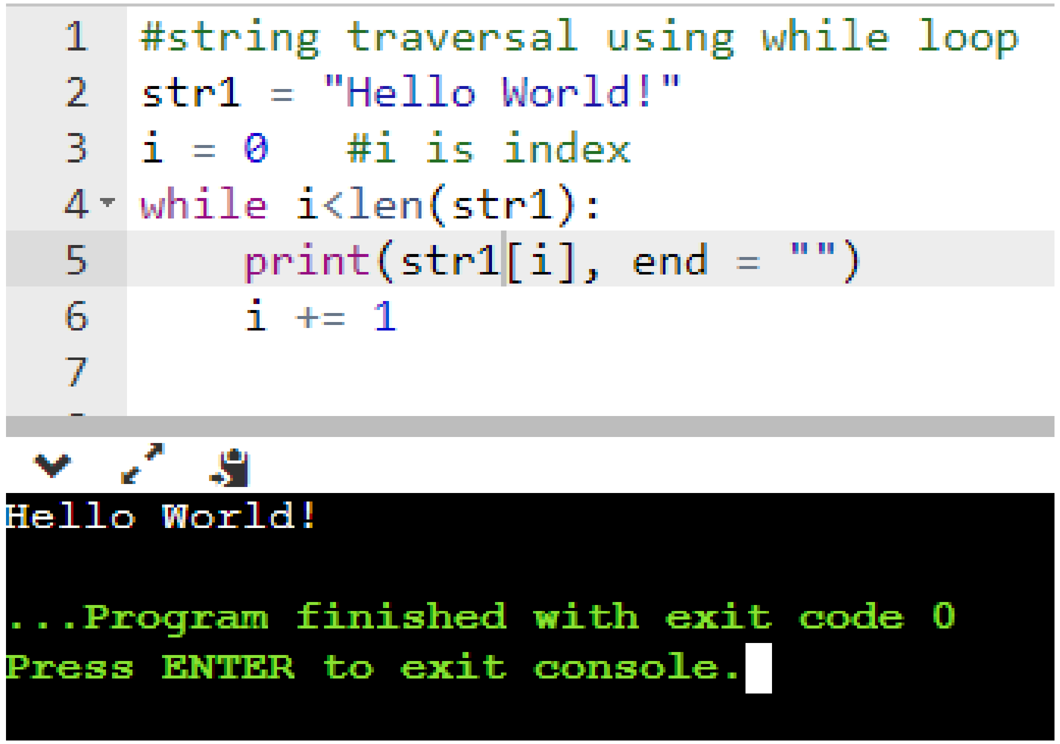
len()	Returns the length of the given string	`>>> str1 = 'Hello World!'`
		`>>> len(str1)`
		`12`

Traversing a string



```python
#string traversal using for loop
str1 = "Hello World!"
for i in str1:
    print(i, end="")

```

```
Hello World!

...Program finished with exit code 0
Press ENTER to exit console.
```



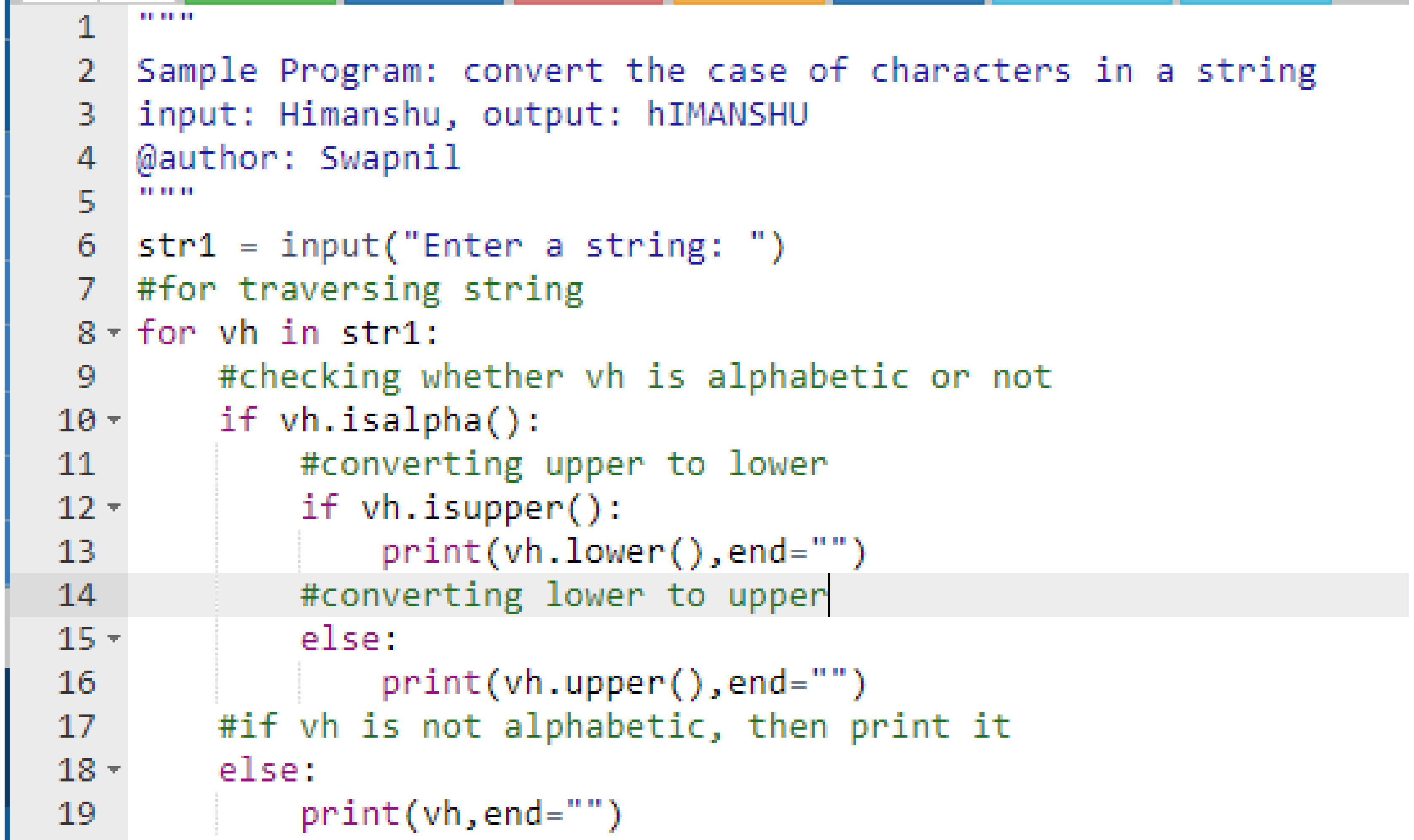
```python
#string traversal using while loop
str1 = "Hello World!"
i = 0   #i is index
while i<len(str1):
    print(str1[i], end = "")
    i += 1

```

```
Hello World!

...Program finished with exit code 0
Press ENTER to exit console.
```

Programming Problems on Strings

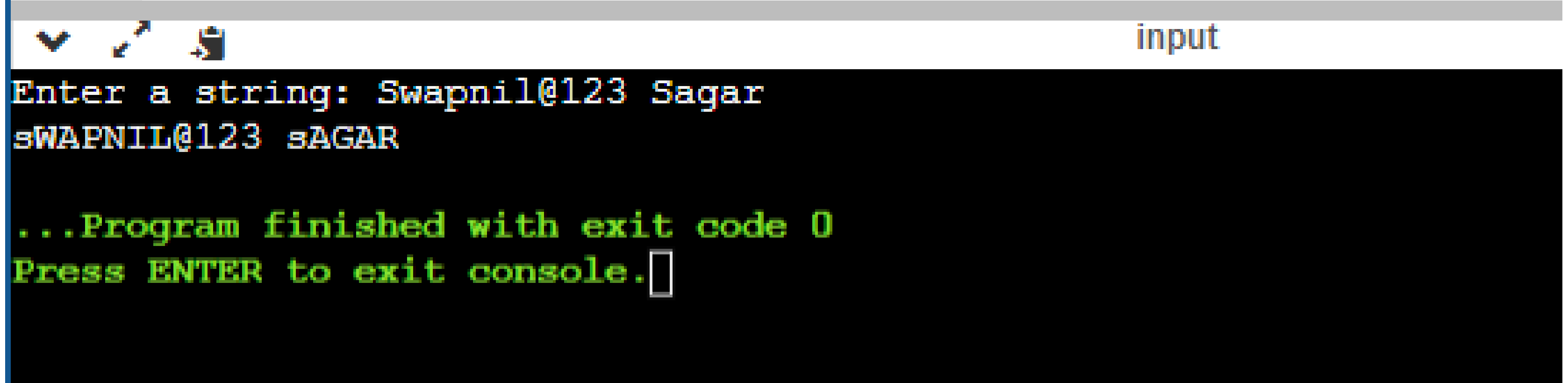
- Sample Program: convert the case of characters in a string.

Code:



```
"""
Sample Program: convert the case of characters in a string
input: Himanshu, output: hIMANSHU
@author: Swapnil
"""
str1 = input("Enter a string: ")
#for traversing string
for vh in str1:
    #checking whether vh is alphabetic or not
    if vh.isalpha():
        #converting upper to lower
        if vh.isupper():
            print(vh.lower(),end="")
        #converting lower to upper
        else:
            print(vh.upper(),end="")
    #if vh is not alphabetic, then print it
    else:
        print(vh,end="")
```

Output:

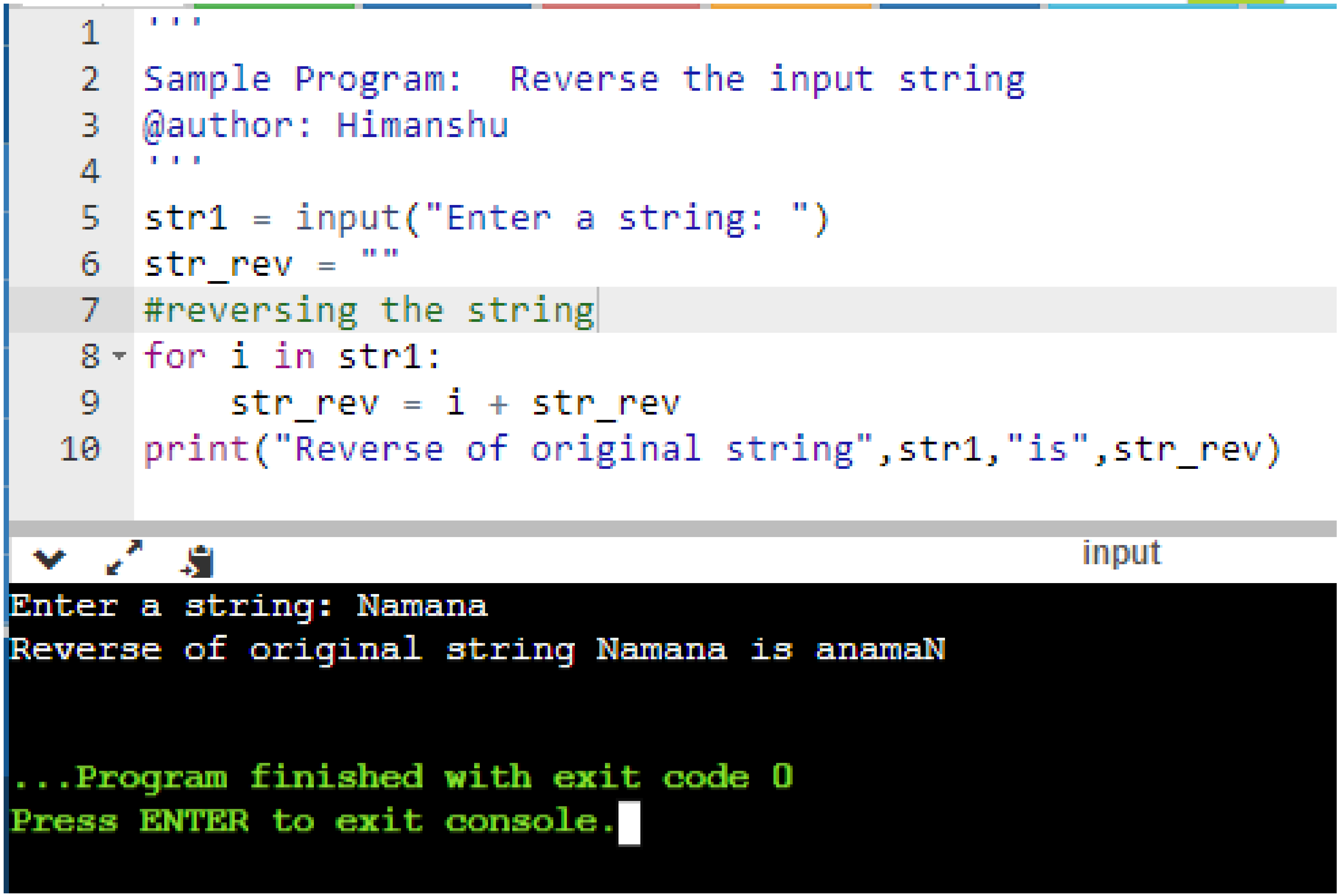


```
Enter a string: Swapnil@123 Sagar
sWAPNIL@123 sAGAR

...Program finished with exit code 0
Press ENTER to exit console.
```

Programming Problems on Strings

- Sample Program: reverse the input string

Code:



```python
'''
Sample Program:  Reverse the input string
@author: Himanshu
'''
str1 = input("Enter a string: ")
str_rev = ""
#reversing the string
for i in str1:
    str_rev = i + str_rev
print("Reverse of original string",str1,"is",str_rev)
```

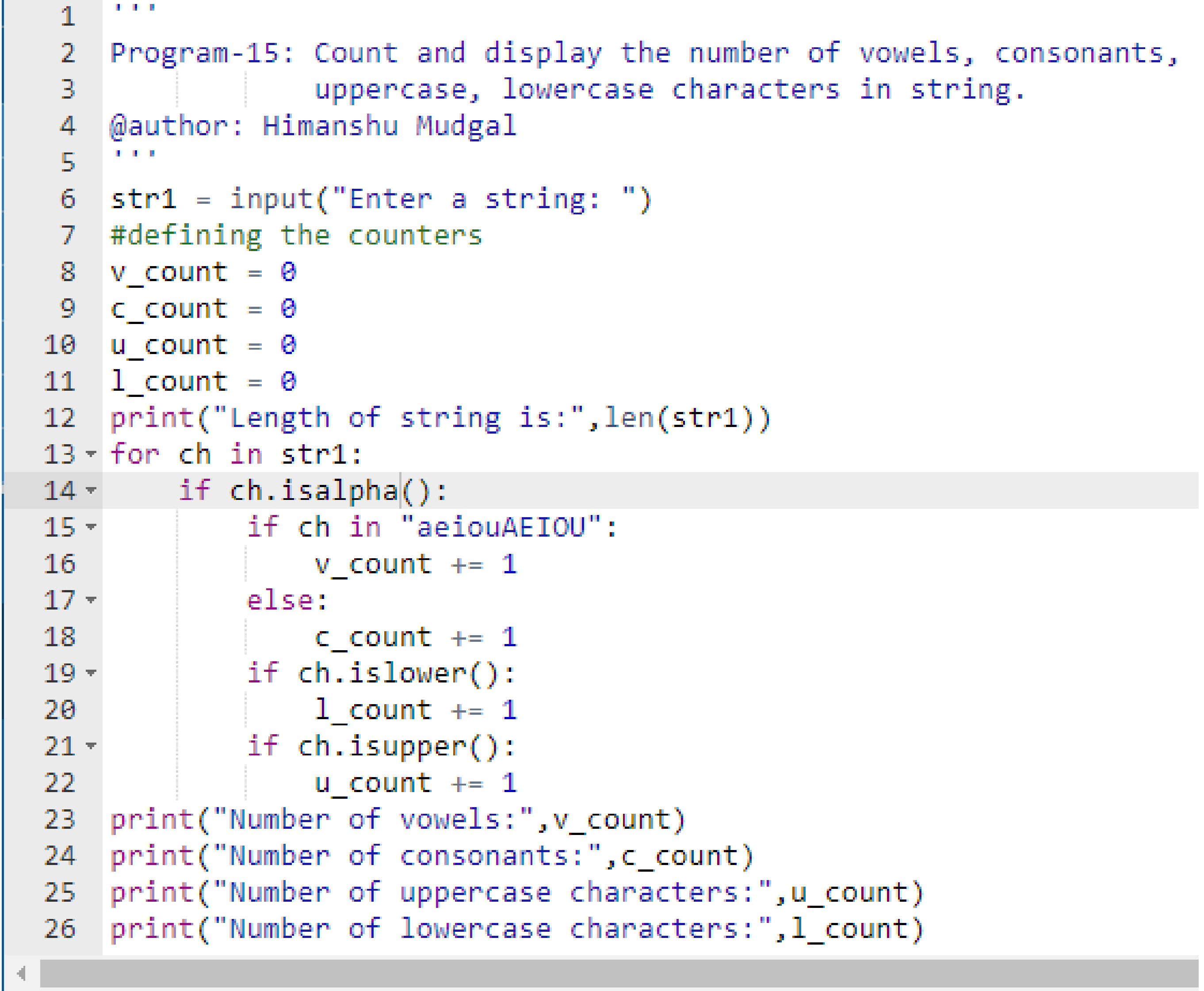
Output:

```
Enter a string: Namana
Reverse of original string Namana is anamaN


...Program finished with exit code 0
Press ENTER to exit console.
```

Programming Problems on Strings

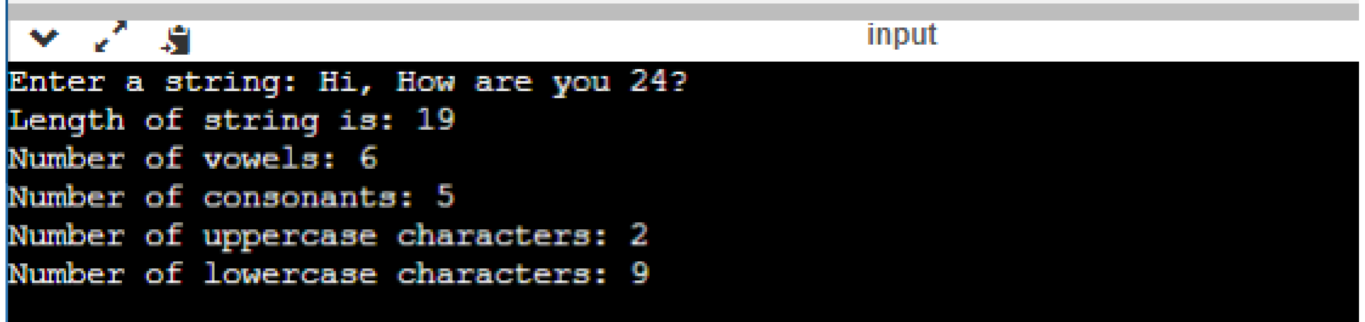
- Program-15: Count and display the number of vowels, consonants, uppercase, lowercase characters in string.

Code:



```python
'''
Program-15: Count and display the number of vowels, consonants,
            uppercase, lowercase characters in string.
@author: Himanshu Mudgal
'''
str1 = input("Enter a string: ")
#defining the counters
v_count = 0
c_count = 0
u_count = 0
l_count = 0
print("Length of string is:",len(str1))
for ch in str1:
    if ch.isalpha():
        if ch in "aeiouAEIOU":
            v_count += 1
        else:
            c_count += 1
        if ch.islower():
            l_count += 1
        if ch.isupper():
            u_count += 1
print("Number of vowels:",v_count)
print("Number of consonants:",c_count)
print("Number of uppercase characters:",u_count)
print("Number of lowercase characters:",l_count)
```

Output:

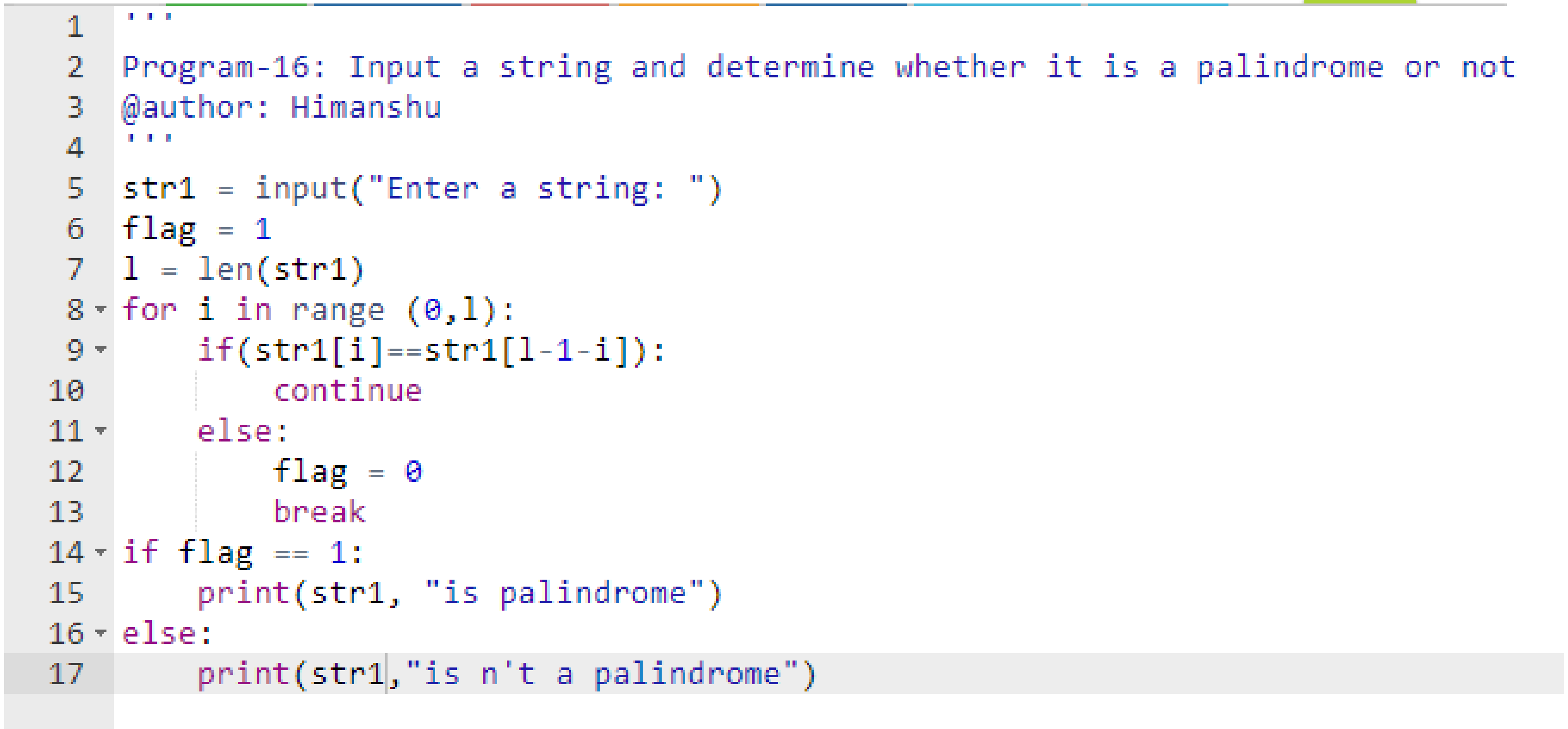


```
Enter a string: Hi, How are you 24?
Length of string is: 19
Number of vowels: 6
Number of consonants: 5
Number of uppercase characters: 2
Number of lowercase characters: 9
```

Programming Problems on Strings

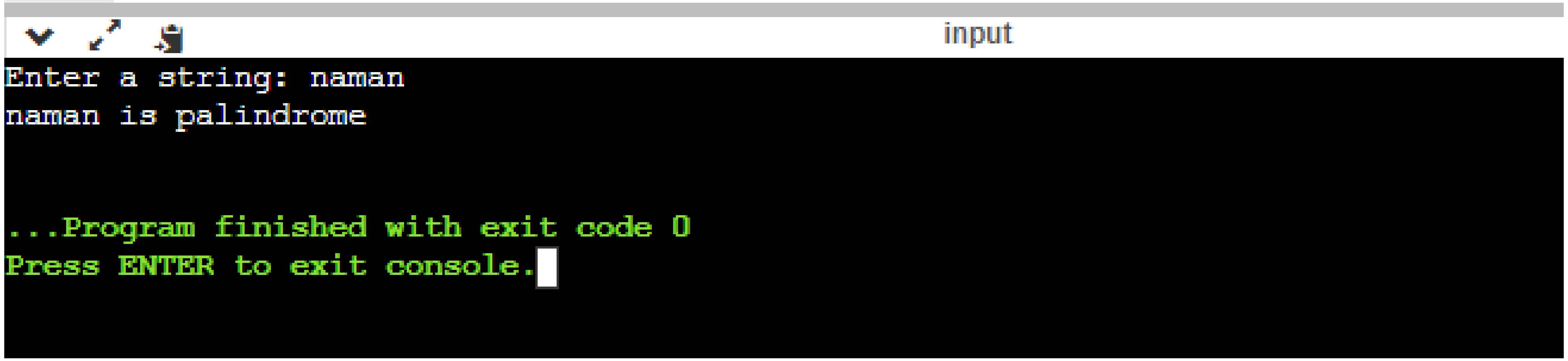
- Program-16: Input a string and determine whether it is a palindrome or not;

Code:



```python
'''
Program-16: Input a string and determine whether it is a palindrome or not
@author: Himanshu
'''
str1 = input("Enter a string: ")
flag = 1
l = len(str1)
for i in range (0,l):
    if(str1[i]==str1[l-1-i]):
        continue
    else:
        flag = 0
        break
if flag == 1:
    print(str1, "is palindrome")
else:
    print(str1,"is n't a palindrome")
```

Output:



```
Enter a string: naman
naman is palindrome


...Program finished with exit code 0
Press ENTER to exit console.
```

Assignment - 3

- Table 8.2: Built-in functions for string manipulations (note down in note book)
- Summary (page-187)

Programming Assignment - 3

- Write a program to count the number of times a character occurs in the given string.
- Write a program which replaces all vowels in the string with '*'.
- Write a program to input a string from the user and print it in the reverse order without creating a new string.
- Write a program to input a string having some digits, and return the sum of digits present in this string.