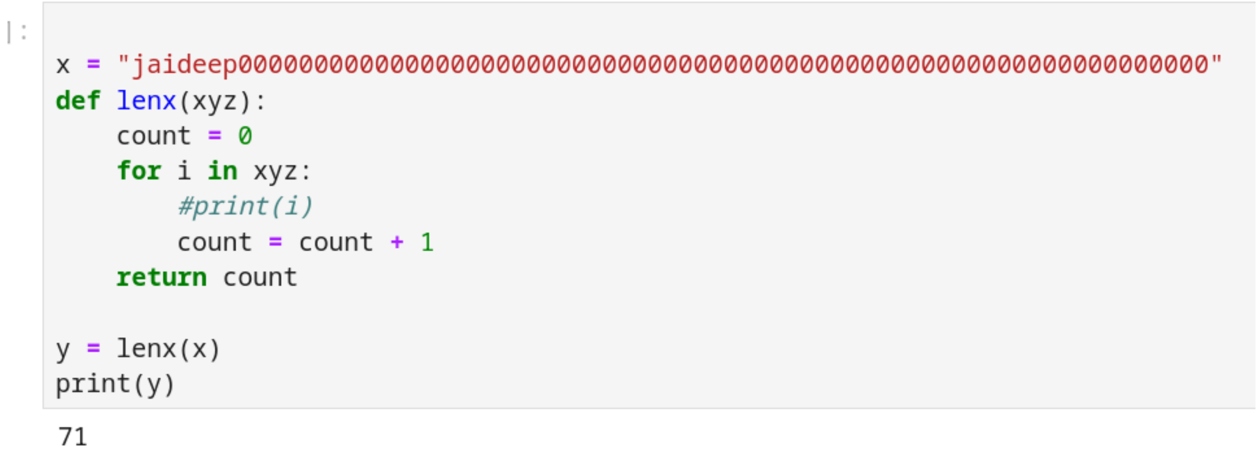
XBit Labs IN - Software Training Institute

code.xbitlabs.in - Free Coding Tutorials

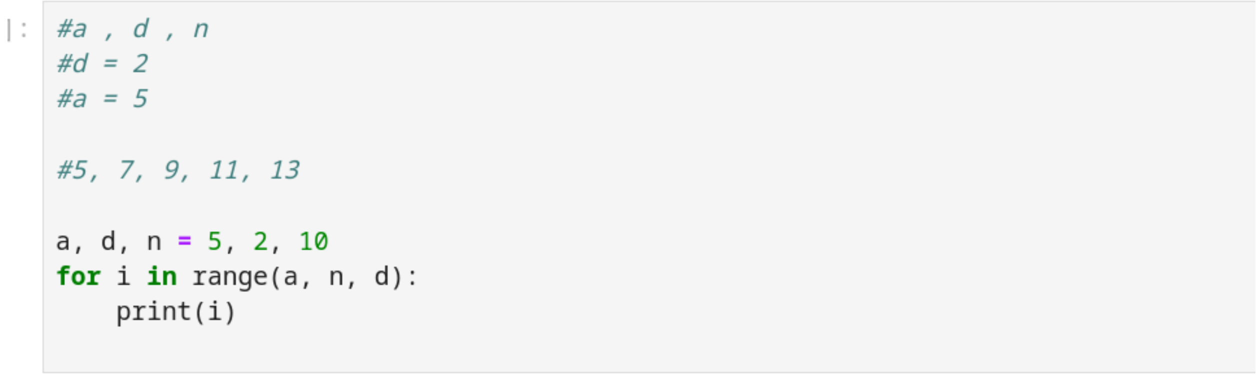
Training Sessions

Date	Aug 3 2024	Session No	2
Subject	Programming/ Problem Solving	Topic	Functions, strings, lists

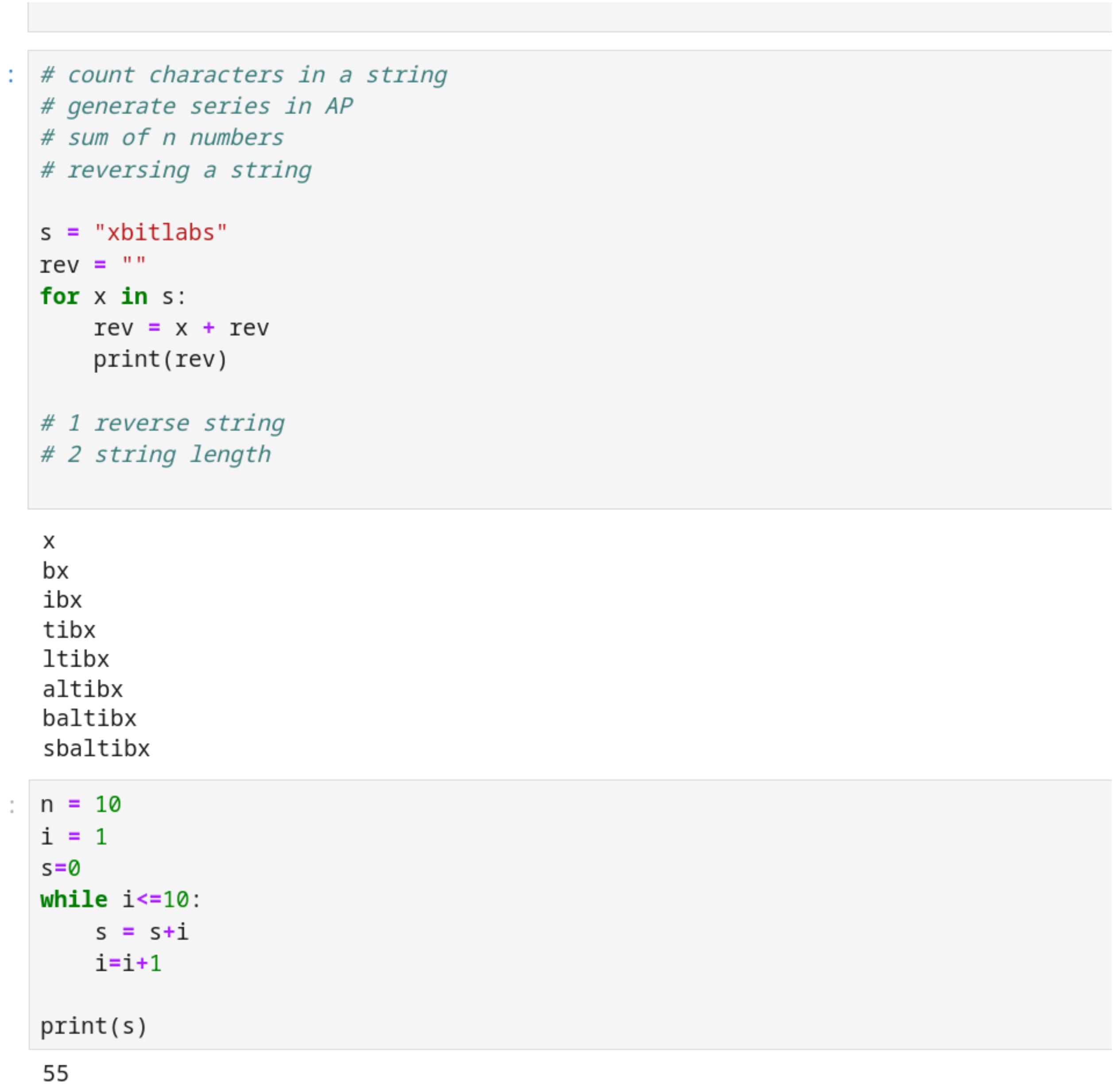


```
x = "jaideep0000000000000000000000000000000000000000000000000000000000000000"
def lenx(xyz):
    count = 0
    for i in xyz:
        #print(i)
        count = count + 1
    return count

y = lenx(x)
print(y)
```

```
71
```



```
#a , d , n
#d = 2
#a = 5

#5, 7, 9, 11, 13

a, d, n = 5, 2, 10
for i in range(a, n, d):
    print(i)
```

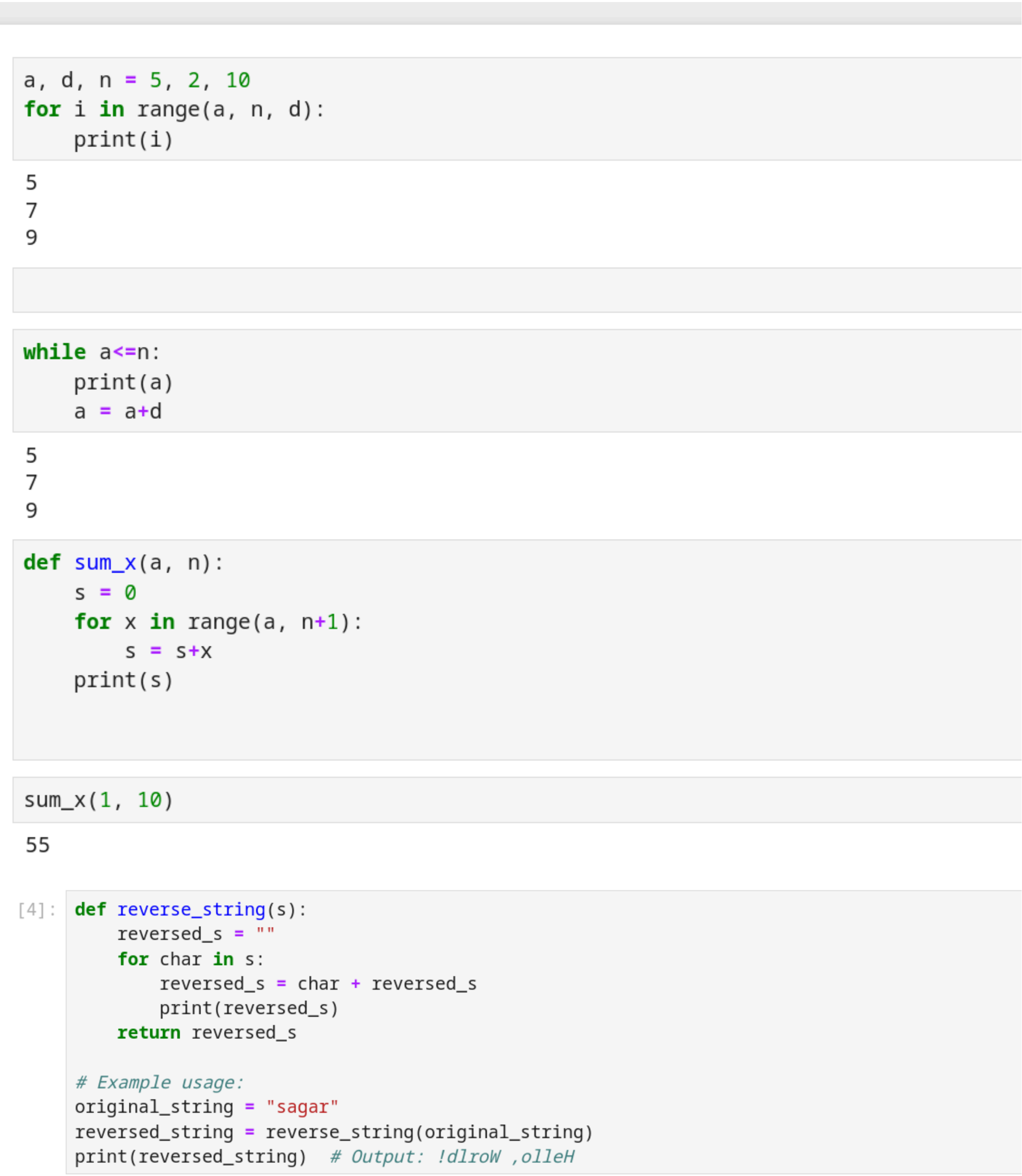


```
# count characters in a string
# generate series in AP
# sum of n numbers
# reversing a string

s = "xbitlabs"
rev = ""
for x in s:
    rev = x + rev
    print(rev)

# 1 reverse string
# 2 string length
```

```
x
bx
ibx
tibx
ltibx
altibx
baltibx
sbaltibx
```

```
n = 10
i = 1
s=0
while i<=10:
    s = s+i
    i=i+1

print(s)
```

```
55
```



```
a, d, n = 5, 2, 10
for i in range(a, n, d):
    print(i)
```

```
5
7
9
```

```
while a<=n:
    print(a)
    a = a+d
```

```
5
7
9
```

```
def sum_x(a, n):
    s = 0
    for x in range(a, n+1):
        s = s+x
    print(s)
```

```
sum_x(1, 10)
```

```
55
```

[4]:

```
def reverse_string(s):
    reversed_s = ""
    for char in s:
        reversed_s = char + reversed_s
        print(reversed_s)
    return reversed_s

# Example usage:
original_string = "sagar"
reversed_string = reverse_string(original_string)
print(reversed_string)  # Output: !dlroW ,olleH
```

[8]:

```python
def sum_of_range(start, end):
    total_sum = 0
    for number in range(start, end + 1):
        total_sum += number
    return total_sum

# Example usage:
start = 1
end = 10
result = sum_of_range(start, end)
print(f"The sum of numbers from {start} to {end} is: {result}")  # Output: The sum of numbers from 1 to 10 is: 55
```

```
The sum of numbers from 1 to 10 is: 55
```

[10]:

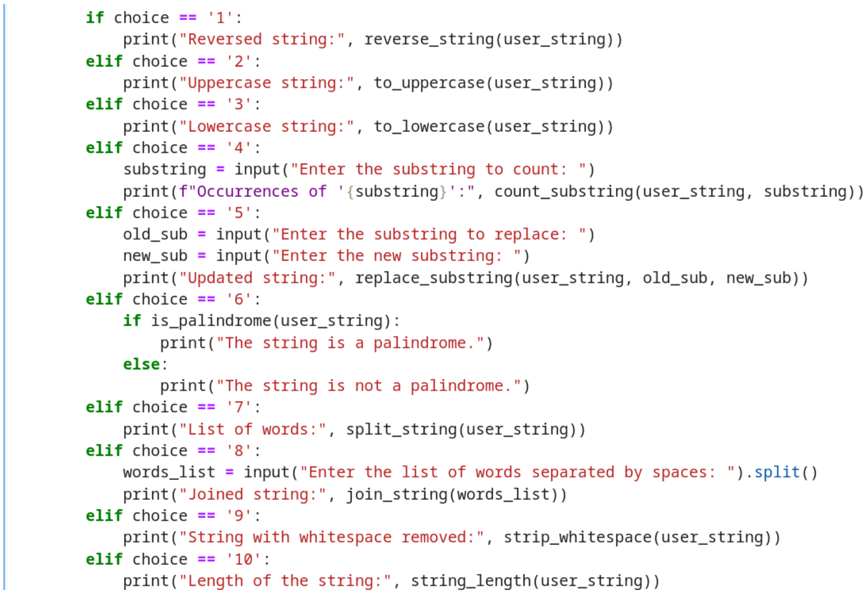
```python
avg = result / (end-start+1)
print(avg)
```

```
5.5
```

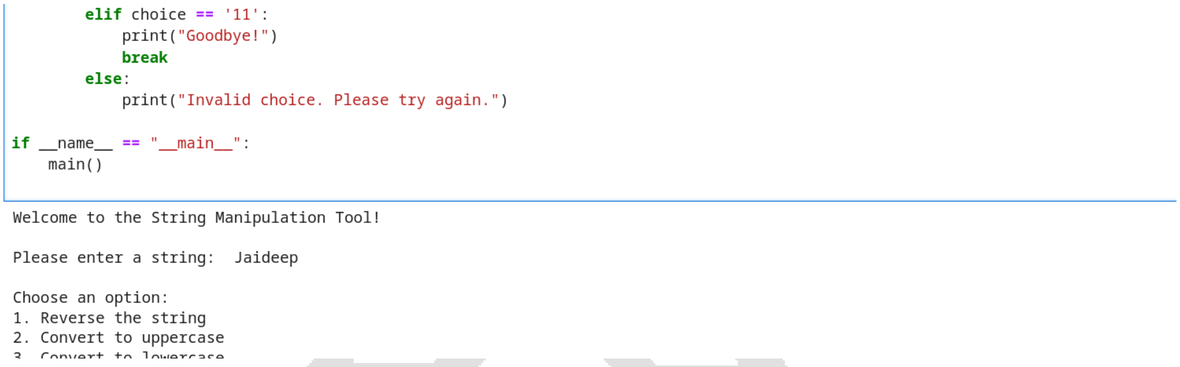
[*]:

```python
def reverse_string(s):
    return s[::-1]

def to_uppercase(s):
    return s.upper()

def to_lowercase(s):
    return s.lower()

def count_substring(s, sub):
    return s.count(sub)

def replace_substring(s, old, new):
    return s.replace(old, new)

def is_palindrome(s):
    return s == s[::-1]

def split_string(s):
    return s.split()

def join_string(lst):
    return ' '.join(lst)

def strip_whitespace(s):
    return s.strip()
```

```python
def strip_whitespace(s):
    return s.strip()

def string_length(s):
    return len(s)

def main():
    print("Welcome to the String Manipulation Tool!")

    user_string = input("\nPlease enter a string: ")

    while True:
        print("\nChoose an option:")
        print("1. Reverse the string")
        print("2. Convert to uppercase")
        print("3. Convert to lowercase")
        print("4. Count occurrences of a substring")
        print("5. Replace a substring with another substring")
        print("6. Check if the string is a palindrome")
        print("7. Split the string into a list of words")
        print("8. Join a list of words into a single string")
        print("9. Remove leading and trailing whitespace")
        print("10. Find the length of the string")
        print("11. Exit")

        choice = input("\nEnter your choice: ")
```

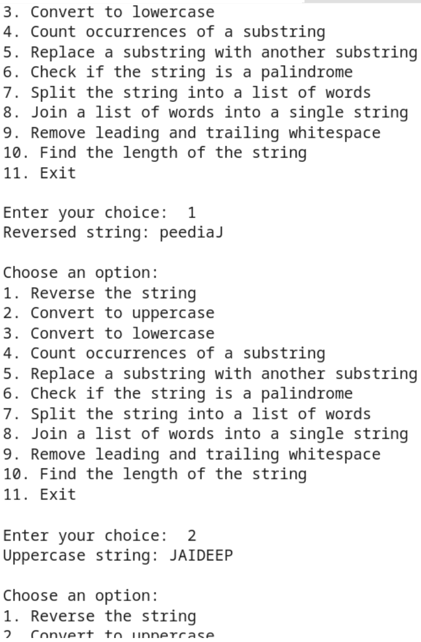


```python
        if choice == '1':
            print("Reversed string:", reverse_string(user_string))
        elif choice == '2':
            print("Uppercase string:", to_uppercase(user_string))
        elif choice == '3':
            print("Lowercase string:", to_lowercase(user_string))
        elif choice == '4':
            substring = input("Enter the substring to count: ")
            print(f"Occurrences of '{substring}':", count_substring(user_string, substring))
        elif choice == '5':
            old_sub = input("Enter the substring to replace: ")
            new_sub = input("Enter the new substring: ")
            print("Updated string:", replace_substring(user_string, old_sub, new_sub))
        elif choice == '6':
            if is_palindrome(user_string):
                print("The string is a palindrome.")
            else:
                print("The string is not a palindrome.")
        elif choice == '7':
            print("List of words:", split_string(user_string))
        elif choice == '8':
            words_list = input("Enter the list of words separated by spaces: ").split()
            print("Joined string:", join_string(words_list))
        elif choice == '9':
            print("String with whitespace removed:", strip_whitespace(user_string))
        elif choice == '10':
            print("Length of the string:", string_length(user_string))
```



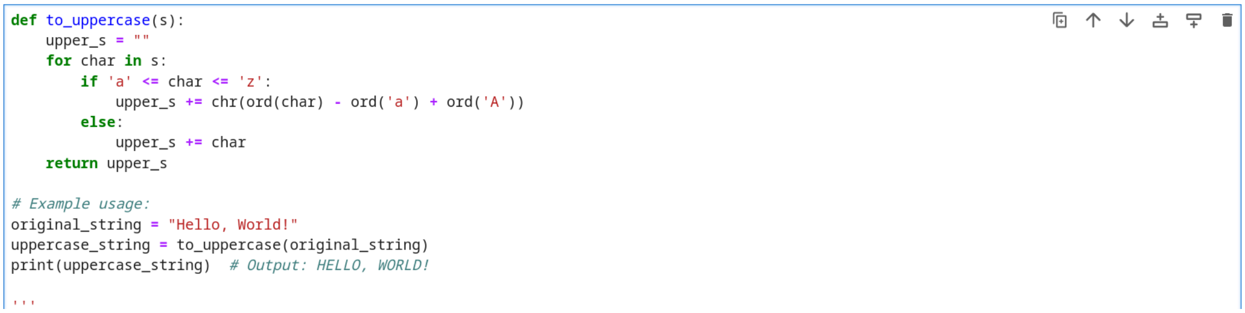
```python
        elif choice == '11':
            print("Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()
```

```
Welcome to the String Manipulation Tool!

Please enter a string:  Jaideep

Choose an option:
1. Reverse the string
2. Convert to uppercase
3. Convert to lowercase
```



```
3. Convert to lowercase
4. Count occurrences of a substring
5. Replace a substring with another substring
6. Check if the string is a palindrome
7. Split the string into a list of words
8. Join a list of words into a single string
9. Remove leading and trailing whitespace
10. Find the length of the string
11. Exit

Enter your choice:  1
Reversed string: peediaJ

Choose an option:
1. Reverse the string
2. Convert to uppercase
3. Convert to lowercase
4. Count occurrences of a substring
5. Replace a substring with another substring
6. Check if the string is a palindrome
7. Split the string into a list of words
8. Join a list of words into a single string
9. Remove leading and trailing whitespace
10. Find the length of the string
11. Exit

Enter your choice:  2
Uppercase string: JAIDEEP

Choose an option:
1. Reverse the string
2. Convert to uppercase
```

Assignment to do :
Convert a string to uppercase, create a function
Reverse a string, create a function



```python
def to_uppercase(s):
    upper_s = ""
    for char in s:
        if 'a' <= char <= 'z':
            upper_s += chr(ord(char) - ord('a') + ord('A'))
        else:
            upper_s += char
    return upper_s

# Example usage:
original_string = "Hello, World!"
uppercase_string = to_uppercase(original_string)
print(uppercase_string)  # Output: HELLO, WORLD!

'''
```

'''

Below is the ASCII (American Standard Code for Information Interchange) table that shows the decimal values, hexadecimal values, and characters for each code from 0 to 127:

Decimal	Hexadecimal	Character	Description
0	0x00	NUL	Null character
1	0x01	SOH	Start of Header
2	0x02	STX	Start of Text
3	0x03	ETX	End of Text
4	0x04	EOT	End of Transmission
5	0x05	ENQ	Enquiry
6	0x06	ACK	Acknowledge
7	0x07	BEL	Bell
8	0x08	BS	Backspace
9	0x09	TAB	Horizontal Tab
10	0x0A	LF	Line Feed
11	0x0B	VT	Vertical Tab
12	0x0C	FF	Form Feed
13	0x0D	CR	Carriage Return
14	0x0E	SO	Shift Out
15	0x0F	SI	Shift In
16	0x10	DLE	Data Link Escape
17	0x11	DC1	Device Control 1
18	0x12	DC2	Device Control 2
19	0x13	DC3	Device Control 3
20	0x14	DC4	Device Control 4
21	0x15	NAK	Negative Acknowledge
22	0x16	SYN	Synchronous Idle
23	0x17	ETB	End of Transmission Block

24	0x18	CAN	Cancel
25	0x19	EM	End of Medium
26	0x1A	SUB	Substitute
27	0x1B	ESC	Escape
28	0x1C	FS	File Separator
29	0x1D	GS	Group Separator
30	0x1E	RS	Record Separator
31	0x1F	US	Unit Separator
32	0x20	(space)	Space
33	0x21	!	Exclamation mark
34	0x22	"	Double quote
35	0x23	#	Number sign (hash)
36	0x24	$	Dollar sign
37	0x25	%	Percent sign
38	0x26	&	Ampersand
39	0x27	'	Single quote
40	0x28	(	Left parenthesis
41	0x29	)	Right parenthesis
42	0x2A	*	Asterisk
43	0x2B	+	Plus sign
44	0x2C	,	Comma
45	0x2D	-	Hyphen (minus)
46	0x2E	.	Period (dot)
47	0x2F	/	Slash (forward slash)
48	0x30	0	Digit 0
49	0x31	1	Digit 1
50	0x32	2	Digit 2
51	0x33	3	Digit 3
52	0x34	4	Digit 4
53	0x35	5	Digit 5
54	0x36	6	Digit 6
55	0x37	7	Digit 7
56	0x38	8	Digit 8
57	0x39	9	Digit 9
58	0x3A	:	Colon
59	0x3B	;	Semicolon
60	0x3C	<	Less-than sign
61	0x3D	=	Equals sign
62	0x3E	>	Greater-than sign
63	0x3F	?	Question mark
64	0x40	@	At sign
65	0x41	A	Uppercase A
66	0x42	B	Uppercase B
67	0x43	C	Uppercase C

68	0x44	D	Uppercase D
69	0x45	E	Uppercase E
70	0x46	F	Uppercase F
71	0x47	G	Uppercase G
72	0x48	H	Uppercase H
73	0x49	I	Uppercase I
74	0x4A	J	Uppercase J
75	0x4B	K	Uppercase K
76	0x4C	L	Uppercase L
77	0x4D	M	Uppercase M
78	0x4E	N	Uppercase N
79	0x4F	O	Uppercase O
80	0x50	P	Uppercase P
81	0x51	Q	Uppercase Q
82	0x52	R	Uppercase R
83	0x53	S	Uppercase S
84	0x54	T	Uppercase T
85	0x55	U	Uppercase U
86	0x56	V	Uppercase V
87	0x57	W	Uppercase W
88	0x58	X	Uppercase X
89	0x59	Y	Uppercase Y
90	0x5A	Z	Uppercase Z
91	0x5B	[	Left square bracket
92	0x5C	\	Backslash
93	0x5D	]	Right square bracket
94	0x5E	^	Caret (circumflex accent)
95	0x5F	_	Underscore
96	0x60	`	Grave accent
97	0x61	a	Lowercase a
98	0x62	b	Lowercase b
99	0x63	c	Lowercase c
100	0x64	d	Lowercase d
101	0x65	e	Lowercase e
102	0x66	f	Lowercase f
103	0x67	g	Lowercase g
104	0x68	h	Lowercase h
105	0x69	i	Lowercase i
106	0x6A	j	Lowercase j
107	0x6B	k	Lowercase k
108	0x6C	l	Lowercase l
109	0x6D	m	Lowercase m
110	0x6E	n	Lowercase n
111	0x6F	o	Lowercase o

112	0x70	p	Lowercase p
113	0x71	q	Lowercase q
114	0x72	r	Lowercase r
115	0x73	s	Lowercase s
116	0x74	t	Lowercase t
117	0x75	u	Lowercase u
118	0x76	v	Lowercase v
119	0x77	w	Lowercase w
120	0x78	x	Lowercase x
121	0x79	y	Lowercase y
122	0x7A	z	Lowercase z
123	0x7B	{	Left curly brace
124	0x7C		
125	0x7D	}	Right curly brace
126	0x7E	~	Tilde
127	0x7F	DEL	Delete

ASCII Value Usage

Control Characters: ASCII values 0-31 and 127 are control characters (e.g., newline, tab) or formatting characters (e.g., carriage return).

Printable Characters: ASCII values 32-126 are printable characters, including letters, digits, punctuation, and symbols.

'''

END