**Python Programming Basic Assignment-23**

**Question 1**

Create a function that takes a number as an argument and returns True or False depending on whether the number is symmetrical or not. A number is symmetrical when it is the same as its reverse.

Examples

is\_symmetrical(7227) ➞ True

is\_symmetrical(12567) ➞ False

is\_symmetrical(44444444) ➞ True

is\_symmetrical(9939) ➞ False

is\_symmetrical(1112111) ➞ True

In [38]:

**def** is\_symmetrical(num):

**try**:

num**=**str(num)

**if** num**==**num[::**-**1]:

**return** **True**

**else**:

**return** **False**

**except** Exception **as** e:

print(f'Error is: {e}')

In [39]:

is\_symmetrical(7227)

Out[39]:

True

In [40]:

is\_symmetrical(12567)

Out[40]:

False

In [41]:

is\_symmetrical(44444444)

Out[41]:

True

In [42]:

is\_symmetrical(9939)

Out[42]:

False

In [43]:

is\_symmetrical(1112111)

Out[43]:

True

**Question 2**

Given a string of numbers separated by a comma and space, return the product of the numbers.

Examples

multiply\_nums("2, 3") ➞ 6

multiply\_nums("1, 2, 3, 4") ➞ 24

multiply\_nums("54, 75, 453, 0") ➞ 0

multiply\_nums("10, -2") ➞ -20

In [20]:

**def** multiply\_nums(num):

num\_list**=**num**.**split(',')

mul**=**1

**for** i **in** num\_list:

mul**=**mul**\***int(i)

**return** mul

In [21]:

multiply\_nums("2, 3")

Out[21]:

6

In [25]:

multiply\_nums("1, 2, 3, 4")

Out[25]:

24

In [26]:

multiply\_nums("54, 75, 453, 0")

Out[26]:

0

In [27]:

multiply\_nums("10, -2")

Out[27]:

-20

**Question 3**

Create a function that squares every digit of a number.

Examples

square\_digits(9119) ➞ 811181

square\_digits(2483) ➞ 416649

square\_digits(3212) ➞ 9414

Notes

The function receives an integer and must return an integer.

In [72]:

**def** square\_digits(num):

temp**=**[]

**try**:

**while** num**>**0:

rem**=**num**%10**

temp**.**append(str(rem**\*\***2))

num**=**num**//**10

sq**=**temp[::**-**1]

sq**=**''**.**join(sq)

**return** int(sq)

**except** Exception **as** e:

print(f'Exception is: {e}')

In [73]:

square\_digits(9119)

Out[73]:

811181

In [74]:

square\_digits(2483)

Out[74]:

416649

In [75]:

square\_digits(3212)

Out[75]:

9414

**Question 4**

Create a function that sorts a list and removes all duplicate items from it.

Examples

setify([1, 3, 3, 5, 5]) ➞ [1, 3, 5]

setify([4, 4, 4, 4]) ➞ [4]

setify([5, 7, 8, 9, 10, 15]) ➞ [5, 7, 8, 9, 10, 15]

setify([3, 3, 3, 2, 1]) ➞ [1, 2, 3]

In [80]:

**def** setify(l):

**try**:

l**=**set(l)

**except** Exception **as** e:

print(f'Exception is : {e}')

**else**:

**return** list(l)

In [81]:

setify([1, 3, 3, 5, 5])

Out[81]:

[1, 3, 5]

In [82]:

setify([4, 4, 4, 4])

Out[82]:

[4]

In [83]:

setify([5, 7, 8, 9, 10, 15])

Out[83]:

[5, 7, 8, 9, 10, 15]

In [84]:

setify([3, 3, 3, 2, 1])

Out[84]:

[1, 2, 3]

**Question 5**

Create a function that returns the mean of all digits.

Examples

mean(42) ➞ 3

mean(12345) ➞ 3

mean(666) ➞ 6

Notes

• The mean of all digits is the sum of digits / how many digits there are (e.g. mean of digits in 512 is (5+1+2)/3(number of digits) = 8/3=2).

• The mean will always be an integer.

In [88]:

**def** mean(num):

length**=**len(str(num))

sum**=**0

**while** num**>**0:

rem**=**num**%10**

sum**=**sum**+**rem

num**=**num**//**10

avg**=**sum**/**length

**return** int(avg)

In [89]:

mean(42)

Out[89]:

3

In [90]:

mean(12345)

Out[90]:

3

In [91]:

mean(666)

Out[91]:

6