Assignment 23 Solutions

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 [12]:

```
def is_symmetrical(num):
    currentDigit = reversedDigit = 0
    remainingNum = num
    while(remainingNum != 0):

        currentDigit = remainingNum % 10

        reversedDigit = reversedDigit * 10 + currentDigit
        print('Reveresed Digit :',reversedDigit)
        remainingNum = remainingNum // 10

    if reversedDigit == num:
        print("f'Num {} is symmetrical'.format(num) ➞ {True}")
    else:
        print("f'Num {} is not symmetrical'.format(num) ➞ {Fales}")

is_symmetrical(7227)
is_symmetrical(12567)
is_symmetrical(44444444)
is_symmetrical(9939)
is_symmetrical(1112111)
```

```
Reveresed Digit : 7
Reveresed Digit : 72
Reveresed Digit : 722
Reveresed Digit : 7227
f'Num {} is symmetrical'.format(num) ➞ {True}
Reveresed Digit : 7
Reveresed Digit : 76
Reveresed Digit : 765
Reveresed Digit : 7652
Reveresed Digit : 76521
f'Num {} is not symmetrical'.format(num) ➞ {Fales}
Reveresed Digit : 4
Reveresed Digit : 44
Reveresed Digit : 444
Reveresed Digit : 4444
Reveresed Digit : 44444
Reveresed Digit : 444444
Reveresed Digit : 4444444
Reveresed Digit : 44444444
f'Num {} is symmetrical'.format(num) ➞ {True}
Reveresed Digit : 9
Reveresed Digit : 93
Reveresed Digit : 939
Reveresed Digit : 9399
f'Num {} is not symmetrical'.format(num) ➞ {Fales}
Reveresed Digit : 1
Reveresed Digit : 11
Reveresed Digit : 111
Reveresed Digit : 1112
Reveresed Digit : 11121
Reveresed Digit : 111211
Reveresed Digit : 1112111
f'Num {} is symmetrical'.format(num) ➞ {True}
```

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 [14]:

```
def multiply_nums(s):
    s = s.replace(' ', "")
    s = s.split(',')
    sum = 1
    for i in s:
        sum = sum * int(i)
    return sum

multiply_nums("2, 3")
```

Out[14]:

```
6
```

In [15]:

```
multiply_nums('1, 2, 3, 4')
```

Out[15]:

```
24
```

In [16]:

```
multiply_nums('54, 75, 453, 0')
```

Out[16]:

```
0
```

In [17]:

```
multiply_nums('10, -2')
```

Out[17]:

```
-20
```

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 [20]:

```python
def square_digits(num):
    p = ''.join(str(int(i)**2) for i in str(num))
    return int(p)
square_digits(9119)
```

Out[20]:

```
811181
```

In [21]:

```python
square_digits(2483)
```

Out[21]:

```
416649
```

In [22]:

```python
square_digits(3212)
```

Out[22]:

```
9414
```

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 [23]:

```python
def setify(in_list):
    out_list = sorted(set(in_list))
    print(f'{in_list} ➞ {out_list}')

setify([1, 3, 3, 5, 5])
setify([4, 4, 4, 4])
setify([5, 7, 8, 9, 10, 15])
setify([3, 3, 3, 2, 1])
```

```
[1, 3, 3, 5, 5] ➞ [1, 3, 5]
[4, 4, 4, 4] ➞ [4]
[5, 7, 8, 9, 10, 15] ➞ [5, 7, 8, 9, 10, 15]
[3, 3, 3, 2, 1] ➞ [1, 2, 3]
```

5.Create a function that returns the mean of all digits.

Examples:

```
mean(42) ➞ 3
mean(12345) ➞ 3
mean(666) ➞ 6
```

Notes:

1.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).

2.The mean will always be an integer.

In [24]:

```
def mean(n):
    N = len(str(n))
    sum = mean = 0

    for digit in str(n):
        sum += int(digit)
    return int(sum/N)
mean(42)
```

Out[24]:

```
3
```

In [25]:

```
mean(12345)
```

Out[25]:

```
3
```

In [26]:

```
mean(666)
```

Out[26]:

```
6
```