Exercise 1: Create a function that takes in three arguments, two of which are optional. The first argument should be a required positional argument, the second argument should be a keyword argument with a default value of 10, and the third argument should be a keyword argument with a default value of None. The function should print the sum of the first two arguments if the third argument is None, and print the product of all three arguments if the third argument is not None.

In [1]:
```
def x(arg1,arg2=10,arg3=None):
    if arg3 is None:
        ans1=arg1+arg2
        print("arg3 is none",ans1)
    else:
        ans2=arg1*arg2*arg3
        print("arg3 is not none",ans2)
x(5)
x(9,7)
x(3,2,8)
```

```
arg3 is none 15
arg3 is none 16
arg3 is not none 48
```

Exercise 2:

Write a function that takes in a list of strings and returns a new list with only the strings that have a length greater than or equal to 5.

In [2]:
```
list=[]
def x(string):
    for i in string:
        if len(i) >= 5 :
            list.append(i)
    return list
string=["Orange","Pineapple","Grapes","Pappaya","Kiwi"]
list=x(string)
print(list)
```

```
['Orange', 'Pineapple', 'Grapes', 'Pappaya']
```

Exercise 3:

Write a Python program to evaluate a given mathematical expression using the eval() function. expression = "3 * 5 + 2"

In [3]:
```
expression="3*5+2"
```

```
ans=eval(expression)
print(ans)
```

```
17
```

Exercise 4:

Write a Python program to filter out the prime numbers from a given list of integers using the filter() function.

In [15]:
```
numbers = [1,2,3,4,5,6,7,8,9,10,11,12]
def prime(num):
    if number <= 1:
        return False
    for i in range(2, int(number**0.5) + 1):
        if number % i == 0:
            return False
    return True

x= list(filter(prime, numbers))
print(x)
```

```
[2, 3, 5, 7, 11]
```

Exercise 5:

Write a Python program to convert a list of strings to uppercase using the map() function.

In [5]:
```
string = ["Apple", "Orange", "Mango", "Kiwi"]
a = [s.upper() for s in string]
print(a)
```

```
['APPLE', 'ORANGE', 'MANGO', 'KIWI']
```

Exercise 6:

Write a Python program to calculate the length of each string in a given list of strings using the map() function.

In [6]:
```
x=["Apple","Orange","Mango","Kiwi"]
y=list(map(len,x))
print(y)
```

```
[5, 6, 5, 4]
```

Exercise 7:

Write a Python program to calculate the sum of elements in a list using the reduce() function.

In [7]:
```
from functools import reduce
x=["Apple","Orange","Mango","Kiwi"]
y= sum(len(s) for s in x)
print(y)
```

```
20
```

Exercise 8:

Write a Python program to find the maximum element in a list using the reduce() function.

In [8]:
```
from functools import reduce

def max_value(arr):
    return reduce(lambda a, b: a if a > b else b, arr)

arr = [3, 5, 1, 2, 9, 5, 4]
n = max_value(arr)
print(n)
```

```
9
```

Exercise 9:

Write a generator function that takes a list of integers and yields only the even numbers

In [9]:
```
def even(num):
    for i in num:
        if i % 2 == 0:
            yield i
x=even([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
for j in x:
    print(j)
```

```
2
4
6
8
10
```

Exercise 10:

Write a decorator function that prints the execution time of a function.

```
In [10]: def xtime(func):
             def lit(*a, **b):
                 x = time.time()
                 z = func(*a, **b)
                 y = time.time()
                 print(f"It takes {y-x:.4f} seconds to execute.")
                 return z
             return lit

         @xtime
         def w(n):
             return sum(i**2 for i in range(n))

         w(10000)
```

```
It takes 0.0020 seconds to execute.
```

```
Out[10]: 333283335000
```