In [ ]:
```
def fseries(n):
    x,y=1,1
    l=[x,y]
    for i in range(2,n):
        temp=x
        x=y
        y=temp+y
        l.append(y)
    return l
print(fseries(5))
print(fseries(6))
```

```
[1, 1, 2, 3, 5]
[1, 1, 2, 3, 5, 8]
```

In [ ]:
```
def rfseries(n,l=[1,1]):
    if n==1:
        return l
    l.append(l[-2]+l[-1])
    return rfseries(n-1,l)
print(rfseries(5))
```

```
[1, 1, 2, 3, 5, 8]
```

In [ ]:
```
lrfseries=lambda n,l=[1,1]:l if n==1 else lrfseries(n-1,l+[l[-2]+l[-1]])
lrfseries(7)
```

Out[ ]:
```
[1, 1, 2, 3, 5, 8, 13, 21]
```

In [ ]:
```
#filter(function,sequence[list,tuple or string])
l=[x for x in range(1,11)]
print(l)

def even(n):
    return n%2==0

def odd(n):
    return n%2!=0

el=list(filter(even,l))
print(el)
et=tuple(filter(odd,l))
print(et)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[2, 4, 6, 8, 10]
(1, 3, 5, 7, 9)
```

In [ ]:
```
l=[x for x in range(1,11)]
el=list(filter(lambda x:x%2==0, l))
print(el)
ot=tuple(filter(lambda x:x%2!=0, l))
print(ot)
```

```
[2, 4, 6, 8, 10]
(1, 3, 5, 7, 9)
```

In [ ]:
```
#map(function, sequence[list,truple,str])
```

```
def doubleit(n):
    return n*2
def squareit(n):
    return n**2

l=[x for x in range(1,11)]
dbll=list(map(doubleit,l))
sql=list(map(squareit,l))
print(l)
print(dbll)
print(sql)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

In [ ]:

```
l=[x for x in range(1,11)]
sql=list(map(lambda x:x**2,l))
dbll=list(map(lambda x:x*2,l))
print(l)
print(dbll)
print(sql)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```