```python
#Python program to display the fibonnaci Sequence
def recur_fibo(n):
    if n<=1:
        return n
    else:
        return(recur_fibo(n-1)+recur_fibo(n-2))
nterms=7
#check if the number of terms is valid
if nterms<=0:
    print("Please enter a positive number")
else:
    print("Fibonacci Sequence:")
    for i in range(nterms):
        print(recur_fibo(i))
```

Output :

```
Fibonacci Sequence:

0

1

1

2

3

5

8
```

```python
#Program to display the fibonacci sequenceup to n th term
nterms=int(input("How many Tearms?"))

#First Two Number
n1=0
n2=1
count=0

#check if the number of terms is valid
if nterms<=0:

    print("Please enter a positive number")

#if there is only one term, return nl
elif nterms==1:
    print("Fibonacci sequence upto",nterms,":")
    print(n1)

#Generate Fibonacci sequence
else:
    print("Fibonacci Sequence:")
    while count<nterms:
        print(n1)
        nth=n1+n2
        #update values
        n1=n2
        n2=nth
        count+=1
```

Output:

```
How many Tearms?5

Fibonacci Sequence:

0

1

1

2

3
```

```
class Item:
    def __init__(self,value,weight):
        self.value=value
        self.weight=weight

def fractionalKnapsack(W,arr):
    #Sorting Items On basis Of Ratio
    arr.sort(key=lambda x:(x.value/x.weight),reverse=True)

    #Reasult(value in Knapsack)
    finalvalue=0.0

    #Looping through all Items
    for item in arr:

        #If adding Items wont overflow,
        #ADD IT COMPLRTELY

        if item.weight<=W:
            W-=item.weight
            finalvalue+=item.value


        #If We cant add curent Item,
        #Add Fractinal part of it
        else:
            finalvalue+=item.value*W/item.weight
            break
    #returning Final Value
    return finalvalue

#Driver Code
if __name__=="__main__":
    W=50
    arr=[Item(60,10),Item(100,20),Item(120,30)]

    #Function Call
    max_val=fractionalKnapsack(W,arr)
    print(max_val)
```

```
output:


240.0
```