

Project

copy the program recursive.py
fact non recursive.py
fact recursive.py
fib non recursive.py
fib recursive.py
floyds.py
gcd non recursive.py
gcd recursive.py
hamiltonian.py
knapsack.py
lcm non recursive.py
lcm recursive.py
max and min.py
max non recursive.py
max recursive.py
mergesort.py
MST.py
multiplication non recursive.py
multiplication recursive.py
n-queens.py
optimal BST.py
palindrome non recursive.py
palindrome recursive.py
prime or not non recursive.py
prime or not recursive.py
scratch.py
stresses multiplication.py

recursive.py × prime or not non recursive.py × MST.py × hamiltonian.py × B&B Travelling salesman.py × sum of subsets.py ×

```
1 def sum_of_subset(s, k, rem):
2     x[k]=1
3     if s+my_list[k]==target_sum:
4         list1=[]
5         for i in range(0, k+1):
6             if x[i]==1:
7                 list1.append(my_list[i])
8             print(list1)
9     elif s+my_list[k]+my_list[k+1]<=target_sum:
10        sum_of_subset(s+my_list[k], k+1, rem-my_list[k])
11    if s+rem-my_list[k]>=target_sum and s+my_list[k+1]<=target_sum:
12        x[k]=0
13        sum_of_subset(s, k+1, rem-my_list[k])
14    my_list=[]
15    n=int(input("Enter number of elements"))
16    total=0
17    for i in range(0, n):
18        ele=int(input())
19        my_list.append(ele)
20        total=total+ele
21    my_list.sort()
22    target_sum=int(input("Enter required Sum"))
23    x=[0]*(n+1)
24    sum of subset(0, 0, total)
```

Project

- copy the program recursive.py
- fact non recursive.py
- fact recursive.py
- fib non recursive.py
- fib recursive.py
- floyds.py
- gcd non recursive.py
- gcd recursive.py

sum of subsets

```
22 target_sum=int(input("Enter required Sum"))
23 x=[0]*(n+1)
24 sum_of_subset(0,0,total)
```

C:\Users\kativ\PycharmProjects\pythonProject2\venv\Scripts\python.exe "C:/Users/kativ/AppData/Roaming/JetBrains/PyCharmCE2022.1/scratches/sum

Enter number of elements5

1

2

3

4

5

Enter required Sum5

[1, 4]

[2, 3]

[5]

Process finished with exit code 0

|