

Write a Python to implement Perfect Sum Problem

```
def sumSubsets(sets, n, target) :  
    x = [0]*len(sets)  
    j = len(sets) - 1  
    while (n > 0) :  
        x[j] = n % 2  
        n = n // 2  
        j -= 1  
    sum = 0  
    for i in range(len(sets)) :  
        if (x[i] == 1) :  
            sum += sets[i]  
    if (sum == target) :  
  
        print("{",end="");  
        for i in range(len(sets)) :  
            if (x[i] == 1) :  
                print(sets[i],end= ", ")  
        print("}", " ",end="")
```

```
def findSubsets(arr, K) :
```

```
    x = pow(2, len(arr))
```

```
    for i in range(1, x) :  
        sumSubsets(arr, i, K)
```

```
#if __name__ == "__main__" :
```

```
arr = []  
n=int(input("enter n value\n"))  
print("enter set elements\n")
```

```
for i in range(0,n):  
    e=int(input())  
    arr.append(e)  
K = int(input("enter sum value\n"))  
print("output\n")  
findSubsets(arr, K)
```

The screenshot shows a Google Colaboratory notebook titled 'Untitled20.ipynb'. The notebook contains a Python program that takes an array of numbers and a target sum, then prints all subsets of the array that sum up to the target. The program is as follows:

```
for i in range(0,n):  
    e=int(input())  
    arr.append(e)  
K = int(input("enter sum value\n"))  
print("output\n")  
findSubsets(arr, K)
```

The notebook interface shows the following input and output:

enter n value
6
enter set elements
5
10
12
13
15
18
enter sum value30
{12, 18, }, {5, 12, 13, }, {5, 10, 15, },

The notebook also shows a sidebar with a file explorer and a bottom status bar with system information.