Tug Of War

1. You are given an array of n integers.
2. You have to divide these n integers into 2 subsets such that the difference of sum of two subsets is as minimum as possible.
3. If n is even, both set will contain exactly n/2 elements. If is odd, one set will contain (n-1)/2 and other set will contain (n+1)/2 elements.
3. If it is not possible to divide, then print "-1".
Note -> Check out the question video and write the recursive code as it is intended without changing signature. The judge can't force you but intends you to teach a concept.

Input Format

A number n
n integers

Output Format

Check the sample ouput and question video.

Constraints

1 <= n <= 20
1 <= arr[i] <= 100

Sample Input

6
1
2
3
4
5
6

Sample Output

[1, 3, 6] [2, 4, 5]

```
import sys



def tugOfWar(array):
```

```
    ans = []
    diff = [sys.maxsize]
    helper(array, ans, diff, [], [], 0)
    return ans


def helper(arr, ans, diff, set1, set2, idx):
    if idx == len(arr):
        if len(set1) == len(set2):
            temp = abs(sum(set1) - sum(set2))
            if temp < diff[0]:
                diff[0] = temp
                if len(ans):
                    ans.pop()
                    ans.pop()
                ans.append(set1[:])
                ans.append(set2[:])
        return

    if len(set1) < len(arr) // 2:
        set1.append(arr[idx])
        helper(arr, ans, diff, set1, set2, idx + 1)
        set1.pop()
    if len(set2) < len(arr) // 2:
        set2.append(arr[idx])
        helper(arr, ans, diff, set1, set2, idx + 1)
        set2.pop()

arr = [1, 2, 3, 4]
print(tugOfWar(arr))
```