```python
def find_pair_with_sum(lst, target):
    complement_set = set()

    for i, num in enumerate(lst):
        complement = target - num
        if complement in complement_set:
            return (complement, num)
        else:
            complement_set.add(num)

    return None  # No pair found

# Example usage
my_list = [2, 7, 11, 15, 3, 6]
target_sum = 9
pair = find_pair_with_sum(my_list, target_sum)
if pair:
    print(f"Pair with sum {target_sum}: {pair}")
else:
    print("No pair found with the given sum.")
```

```python
def targetSum (list, sum):
    list_with_idx = [{'val': val, 'idx': idx} for idx, val in enumerate(list)]

    def sort_by_value (e):
        return e['val']

    # sort the list
    list_with_idx.sort(key=sort_by_value)
    i = 0
    j = len(list) - 1

    while(i < j):
        currSum = list_with_idx[i]['val'] + list_with_idx[j]['val']
        if currSum < sum:
            i+=1
        elif currSum > sum:
            j-=1
        else:
            return [list_with_idx[i]['idx'], list_with_idx[j]['idx']]

    return -1

print(targetSum([2, 2, 5], 4))
```

Short way:

```python
def two_sum(nums, target):
    complement_indices = {}

    for i, num in enumerate(nums):
        complement = target - num
        if complement in complement_indices:
            return [complement_indices[complement], i]
        complement_indices[num] = i

    return None

nums = [3, 5, 2, 8, 4]
target = 6
print(two_sum(nums, target))
```