Programming Solutions

1. Recursive Maximum

```python
def rec_max(nums):
    if len(nums) == 1:
        return nums[0]
    return max(nums[0], rec_max(nums[1:]))

print(rec_max([1, 2, 3]))
print(rec_max([5, 1, 9, 4, 1, 7, 3, 0]))
```

2. Sum to K

```python
def sum_helper(cur, sm, K):
    if sm == K:
        print(cur)
        return
    for i in range(1, K + 1):
        if i + sm > K:
            break
        sum_helper(cur + [i], sm + i, K)

def sum_to_k(K):
    sum_helper([], 0, K)

sum_to_k(3)
```

3. Imbalance

```python
def imbalance_helper(nums, i, s1, s2):
    if i == len(nums):
        return abs(s1 - s2)

    return min(imbalance_helper(nums, i + 1, s1 + nums[i], s2),
               imbalance_helper(nums, i + 1, s1, s2 + nums[i]))

```

```
def imbalance(nums):
    return imbalance_helper(nums, 0, 0, 0)

print(imbalance([1, 2, 3, 6]))
print(imbalance([1, 2, 3, 4, 5, 6, 7, 8, 50]))
print(imbalance([30, 19, 44, 31, 57, 11, 39]))
```