```
public class Coding2
{
    public static boolean groupSum(int start,  int[ ] nums, int target){
        if(start >= nums.length){
            return target== 0;
        }
        return groupSum(start+  1,nums,target-nums[start]) || groupSum(start + 1,nums,target);
    }

    public static boolean groupSum6(int start, int[] nums, int target) {
        if(start >= nums.length){
            return target == 0;
        }else if(nums[start] == 6){
            return groupSum6(start + 1,nums,target-nums[start]);
        } else {
            return groupSum6(start + 1,nums,target-nums[start]) ||  groupSum6(start+1,nums,target);
        }
    }

    public static boolean groupNoAdj(int start, int[] nums, int target) {
        if (start>=nums.length){
            return target==0;
        }else {
            return groupNoAdj (start + 2,nums,target-nums[start]) || groupNoAdj
(start+1,nums,target);
        }
    }

    public static boolean groupSum5(int start, int[] nums, int target) {
        if(start >= nums.length){
            return target == 0;
        }else if(nums[start] % 5 == 0){
            if (start<nums.length-1 && nums[start+1]==1){
                return groupSum5(start+2,nums,target-nums[start]);
            }
            return groupSum5(start+1,nums,target-nums[start]);
        } else {
            return groupSum5(start+1,nums,target-nums[start]) || groupSum5(start+1,nums,target);
        }
    }

    public static boolean splitArray(int[] nums) {
        return helpSplit(0,nums,0,0);
    }

    private static boolean helpSplit(int start,int [] nums,int a, int b){
        if (start>=nums.length){
            return a==b;
        }
        else {
            return helpSplit(start+1,nums,a+nums[start],b) || helpSplit (start+1,nums,a,b+nums[start]);
        }
    }
}
```