John Jane and Chocolates

There are n chocolates in a row, they are numbered from left to right from 1 to n .The size of the ith candy is a(i) .

Jane and John play an interesting and tasty game: they eat candy. Jane will eat candy from left to right, and John — from right to left. The game ends if all the chocolates are eaten.

The process consists of moves. During a move, the player eats one or more sweets from her/his side (Jane eats from the left, John — from the right). Jane makes the first move. During the first move, she will eat 1 candy . Then, each successive move the players alternate — that is, John makes the second move, then Jane, then again John and so on.

On each move, a player counts the total size of chocolates eaten during the current move. Once this number becomes strictly greater than the total size of chocolates eaten by the other player on their previous move, the current player stops eating and the move ends. In other words, on a move, a player eats the smallest possible number of chocolates such that the sum of the sizes of chocolates eaten on this move is strictly greater than the sum of the sizes of chocolates that the other player ate on the previous move. If there are not enough chocolates to make a move this way, then the player eats up all the remaining chocolates and the game ends.

Total sum of sizes of all the chocolates eaten by Jane is: **a**
Total sum of sizes of all the chocolates eaten by John is: **b**

Input:

The first line of input contains T i.e. the number of test cases. The description of T test cases follows.
The first line of each test case contains an integer N denoting number of elements.
The second line of the test case contains an array of sizes of chocolates.

Output:

Every Test case should have an output of 3 integers i.e number of moves required to finish the game,a,b.

Solution:

```
def calculate(arr,n):
   a,b=0,0
   p1=0
   p2=n-1
   ele=n
   toggle=1
   flag,moves=0,0
   prevA,prevB=0,0
   while(ele>0):
          flag=0
       if(toggle==1):
           prevA=0
           moves+=1
           while(flag==0 and ele>0):
               a+=arr[p1]
               prevA+=arr[p1]
               p1+=1
               ele-=1
               if(prevA>prevB):
                   flag=1
       else:
           prevB=0
           moves+=1
           while(flag==0 and ele>0):
               b+=arr[p2]
               prevB+=arr[p2]
               p2-=1
               ele-=1
               if(prevB>prevA):
                   flag=1
       toggle*=-1
   print(moves,a,b)
   return

t=int(input())
while t>0:
   n=int(input())
   arr=list(map(int,input().split()))
   calculate(arr,n)
   t-=1
```