**0 1 2 3 4**

**{2, 7, 1, 8, 4}**

**1. Brute force**

- initialize max = 0

- iterate over the array a with a with index i

- initialize 2 variables sum = i and temp = [i] inside the loop; (in the first loop, sum = 2, temp = [2])

- then create a nested loop with index j = i + 2 and a step of 2 (in the first loop, j = 2, 4)

- append index to the temp list and add value of a[j] to sum (in the first loop, sum = 2+1, temp = [2,1])

- check if sum > max; assign sum to max and temp to ans

case: i = 1, j = 3, sum = 15 which is > max hence max = 15 and ans = [7, 8]

**2. Optimal solution**

- Given that if we take the value at index i, then we cannot take the consecutive index (i + 1) next. So the indexes that we can take are i, i + 2, i+ 4, i + 6, ... making them either odd or even numbers starting from i.

- Hence I created 2 variables, marks\_even and marks\_odd to sum the numbers for odd and even indexes

at i = 0; marks\_even = marks\_even + a[i], and accumulate values at even indexes (j = 2, 4, 6, 8, ...) and store the sum in marks\_even (marks\_even = 2 + 1 + 4)

at i = 1; marks\_odd = marks\_odd + a[i], and accumulate values at even indexes (j = 3, 5, 7, 9, ...) and store the sum in marks\_odd (marks\_odd = 7 + 8)

- After computing the total marks for even and odd indexes, (marks\_even = 7, marks\_odd = 15) we compare them to find the maximum value which in this case is marks\_odd.

- Since marks\_odd is greater, append all the indexes starting 1 (odd is greater) with a step of 2 to the ans array else append from 0