2, 3

4
2 3 -2 4

i = 2 3

max = -12 6

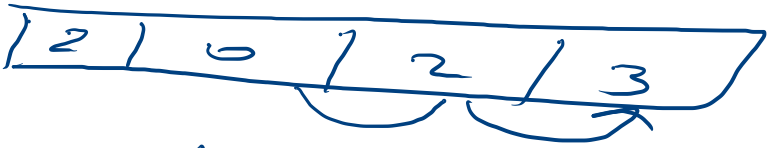
f = 1 * arr[i]

f = 2 x 3 = 6

6 x -2 = -12

-12 x 4 = -48

Expe.ans = 6



i = 0 to 3

max = 0 + 2

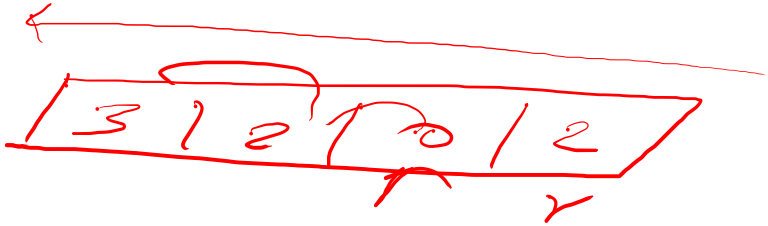
l = 2 × 0 = 0 ⇒ l = (l == 0) ? 1 : left

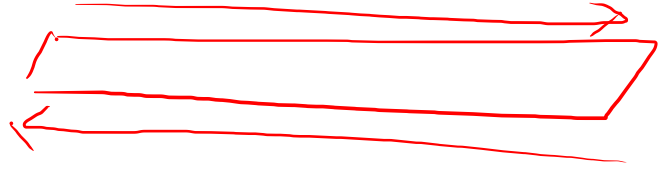
0 × 2 = 0

0 × 3 = 0

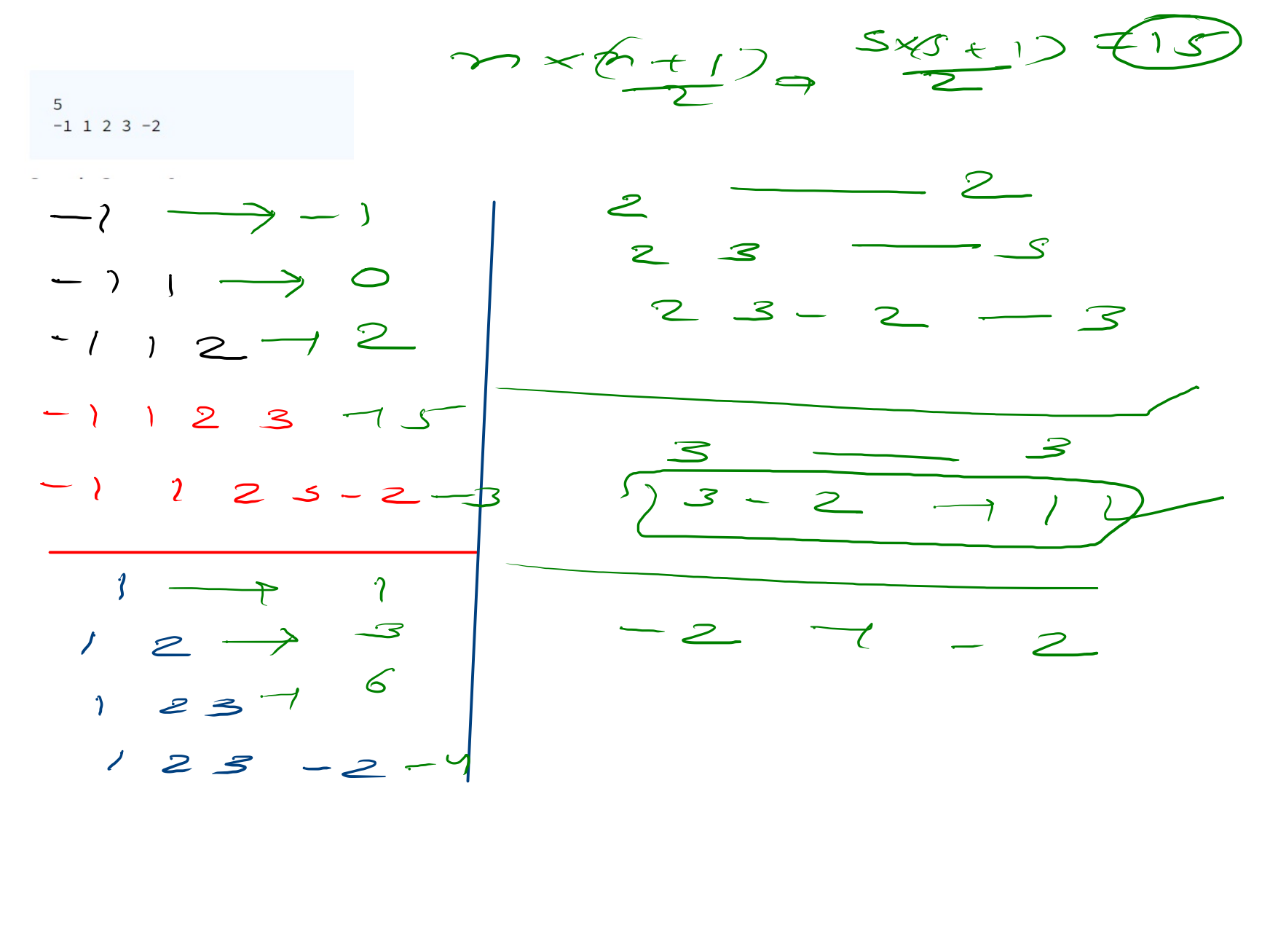
2 × 3

5 × 2

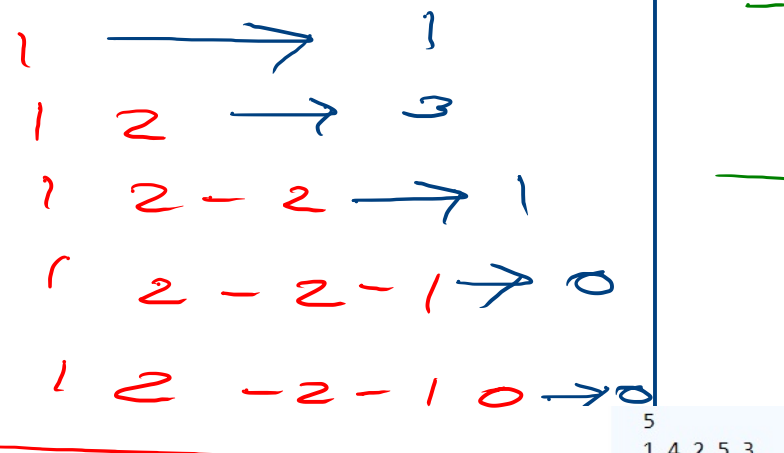




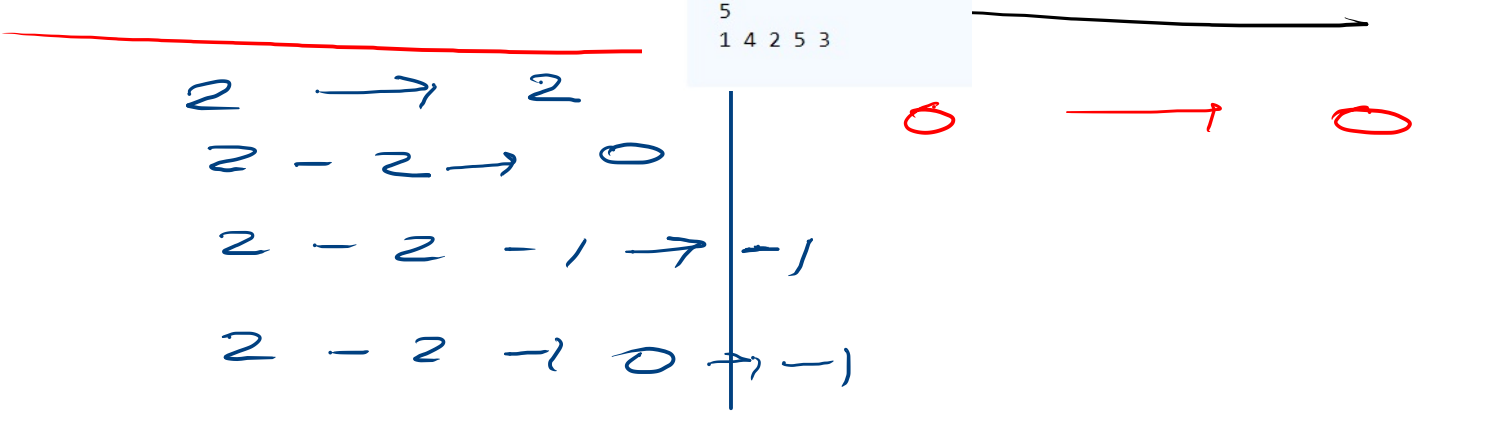
```
5
-1 1 2 3 -2
```

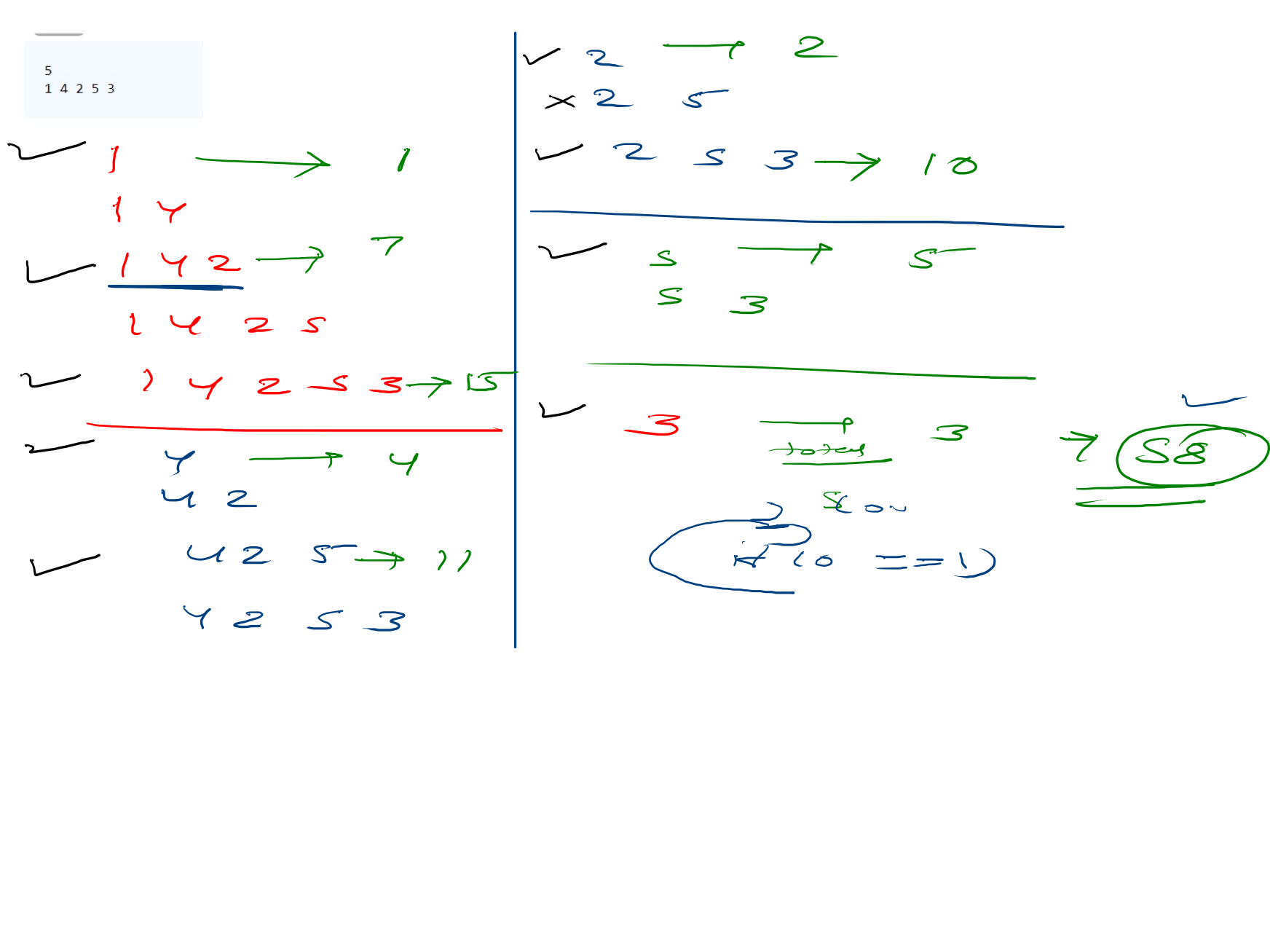

n x (n+1) / 2
5 x (5+1) / 2 = 15
-1 → -1
-1 1 → 0
-1 1 2 → 2
-1 1 2 3 → 5
-1 1 2 3 -2 → 3
1 → 1
1 2 → 3
1 2 3 → 6
1 2 3 -2 → 4
2 → 2
2 3 → 5
2 3 -2 → 3
3 → 3
3 -2 → 1
-2 → -2

```
5
1 2 -2 -1 0
```



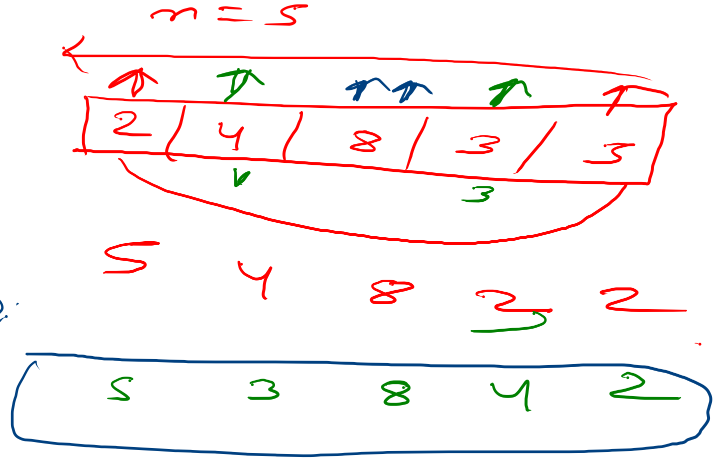
```
5
1 4 2 5 3
```



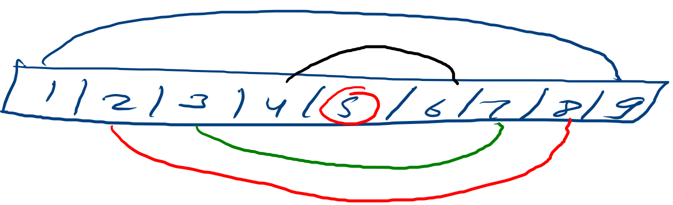

5
1 4 2 5 3
1 → 1
1 4
1 4 2 → 7
1 4 2 5
1 4 2 5 3 → 15
4 → 4
4 2
4 2 5 → 11
4 2 5 3
2 → 2
x 2 5
2 5 3 → 10
5 → 5
5 3
3 → 3
total
58
sum
if (0 == 1)

5
2
4
8
3
5

n-1; i>=0; i--
{
sxso(arr[i]);



2 <= 2 x



```
6
2 5 1 3 4 7
```

$2N \quad \frac{N}{2} = \frac{6}{2} = 3$

$x_1 = 2$

$x_2 = 5$

$x_3 = 1$

$y_1 = 3$

$y_2 = 4$

$y_3 = 7$

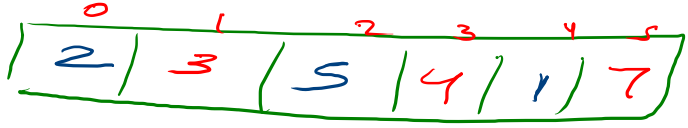
```
[x1, y1, x2, y2, ..., xn, yn].
```

2	3	5	4	1	7

6
2 5 1 3 4 7

for(int i=0; i<n/2; i++)

count=0 2 4



count=1 3 5 7

for(int i=n/2; i<n; i++)

4<6 5<6
3<6 ✗