[4   3   −2   6   −14 { 9   −1   4   5   7   −10   2   9   −10   −5   −9 (6   1)]

max = 0̶ 4̶ 7̶ 11̶ 12̶ 17̶ 24̶ 21̶
c sum = 0̶ 4̶ 7̶ 8̶ 11̶ −3̶ 9̶ 8

12̶ 17̶ 24̶ 19̶ 16̶ 25̶ 17̶

10̶ 11̶ 8

(11)

= 4

4   3

4   3   −2

4   3   −2   6

4   3   −2   6   −14
α α α (−3) =

9 =

9   −1

9   −1   4

9   −1   4   5

9   −1   4   5   7

9   −1   4   5   7   −10

9   −1   4   5   7   −10   2

9   −1   4   5   7   −10   2   9

9   −1   4   5   7   −10   2   9   −10

9   −1   4   5   7   −10   2   9   −10   −5

11 max = 0̶0̶ −2   −9
CSum = 0̶ −2

−2 {−3} −1   −5   −61

1.

$$[\,2 \quad -10 \quad 3 \quad 5 \quad -9 \quad -4 \quad 4 \quad 9 \quad 8 \quad 6 \quad 5\,]$$

8   30
8

2   -8   -68   -300   2700

240

240
9
2160 x

8640

max = 1 x 2700

8640

3 2 7 −6   5 −3 2 −1 −8 3 9 4 2 8

max = 1̶ 3̶ 6̶ 4̶2̶ 3̶7̶8̶0̶ 7560
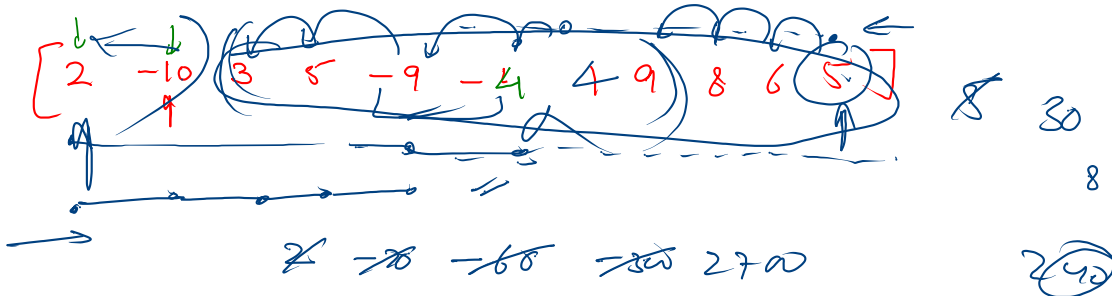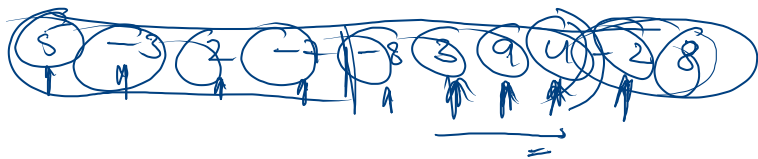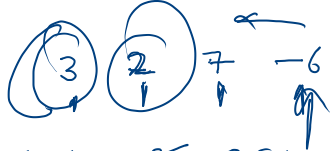
prod = 1̶ 3̶ 6̶ 4̶2̶ −2̶5̶2̶ −1260 3780 7560 −52920

3 2 0 −6 5 −3 0 −7 −8 3 0 4 −2 8
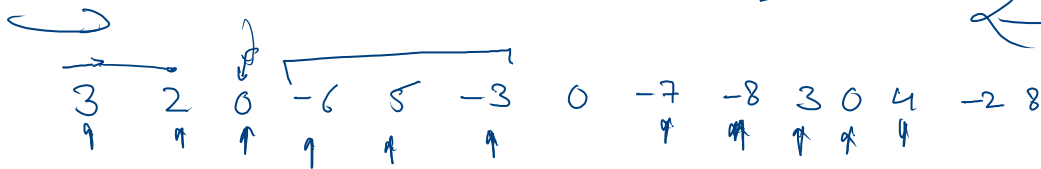
ans = −6̶ 3̶6̶ 9̶0̶ [168]

prod = 1̶ 3̶ 6̶ 0̶ 1̶ 6̶ −30 9̶0̶ 0̶ 1̶ 7̶ 5̶6̶ 1̶6̶8̶0̶ 1̶ 4 −8̶ −64

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for(int i = 0 ; i < n ; i++) arr[i] = scn.nextInt();
    System.out.println(getMaxKadanes(arr));
}

public static int getMaxKadanes(int[] arr) {
    int cSum = 0, maxSum = Integer.MIN_VALUE;
    for(int ele : arr) {
        cSum += ele;
        if(cSum > maxSum) maxSum = cSum;
        if(cSum < 0) cSum = 0;
    }
    return maxSum;
}
```

-1 2 3 -2 1

cSum = 0 -1
maxSum = -∞ -1

(S n - 0)   0

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for(int i = 0 ; i < n ; i++) arr[i] = scn.nextInt();
    System.out.println(getMaxKadanes(arr));
}

public static int getMaxKadanes(int[] arr) {
    int cSum = 0, maxSum = Integer.MIN_VALUE;
    for(int i = 0; i < arr.length; i ++) {
        int ele = arr[i];
        cSum += ele;
        if(cSum > maxSum) maxSum = cSum;
        if(cSum < 0) cSum = 0;
    }
    return maxSum;
}
```