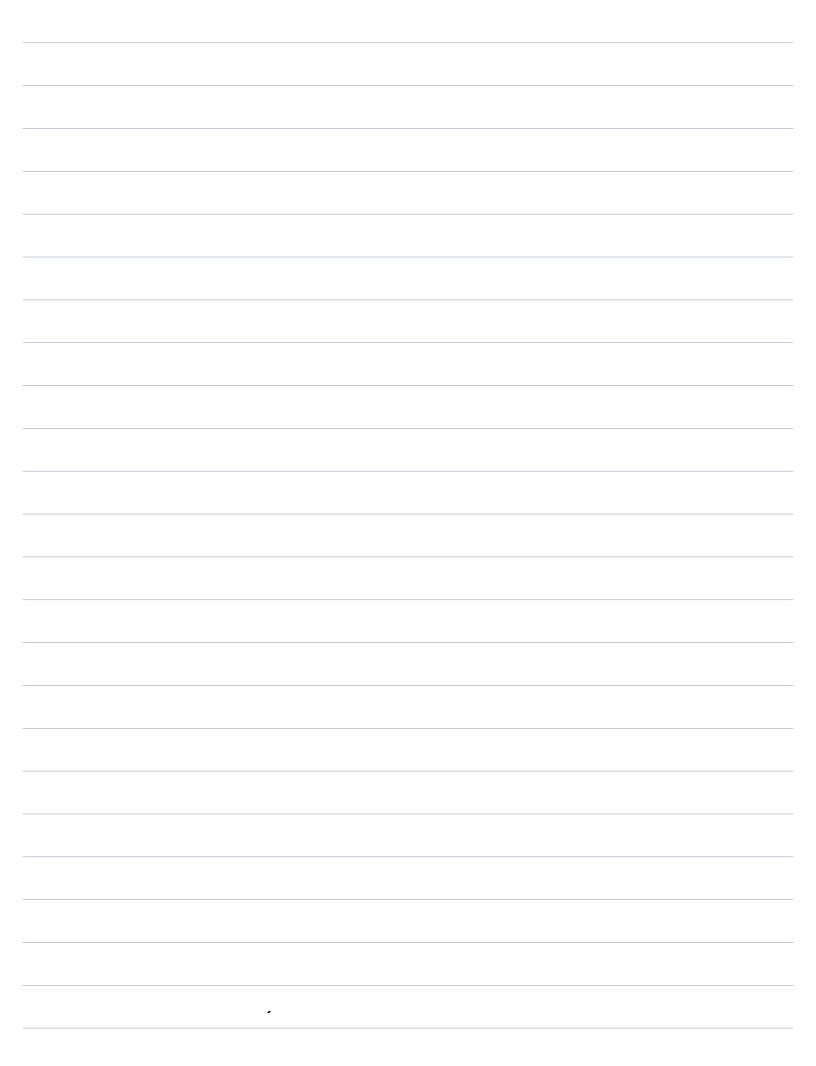
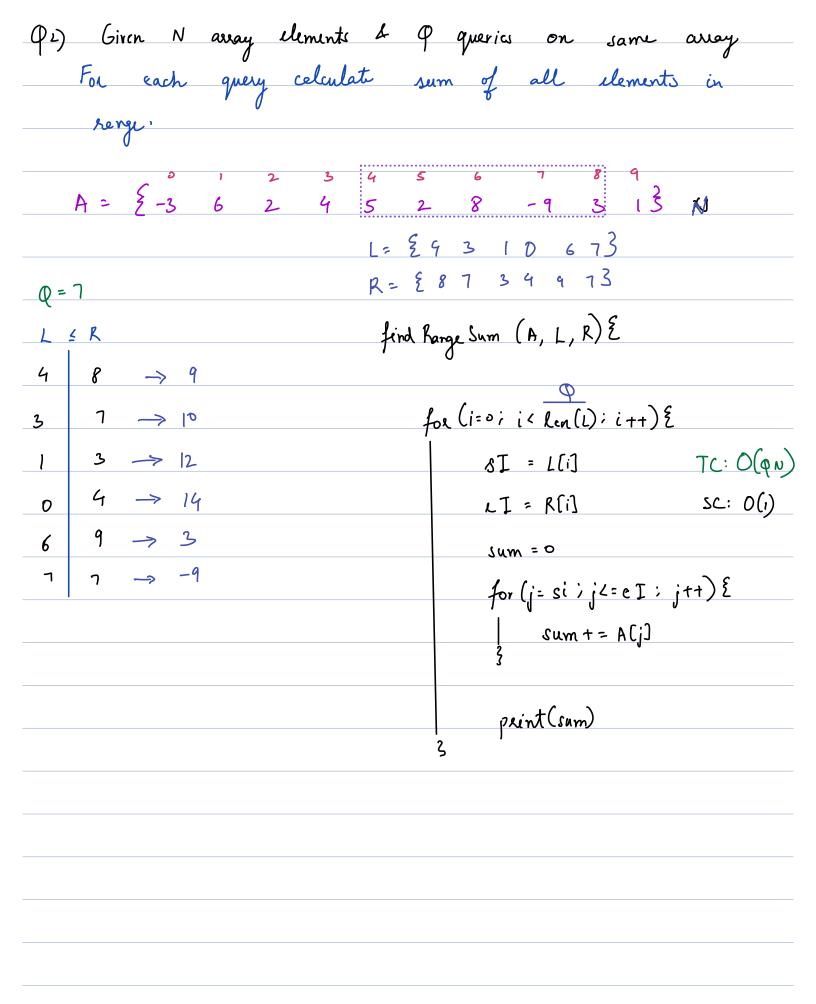
$A = \begin{cases} 5 & 2 & -1 & 0 & 4 & 3 & 2 \\ 3 & 4 & 5 & 6 \\ 4 & 3 & 2 & 3 \end{cases}$ Ans = $\begin{cases} 5 & 7 & 6 & 6 & 10 & 13 & 15 \\ 3 & 6 & 6 & 10 & 13 & 15 \\ 4 & 6 & 6 & 10 & 13$

A: [10 32 6 12 20 1] = \(\frac{10}{10} \) \(\frac{42}{48} \) \(\frac{60}{80} \) \(\frac{80}{3} \)

```
A= \( \frac{2}{3} \) 2 -1 5 2 \( \frac{2}{3} \)
                     pf[3) = A(0) + A(1) + A(2) + A(3)
Normal
             Pt = new Array[N]
                                                    TC: O(N2)
            fox(i=0; i<N;i++){
                                                 3C: D(N)
                 fox (j=0; j<=i; j++) { pf[i]
                   Pfti) = sum
             paint (pf)
```





$$A = \begin{cases} 2 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ -3 & 6 & 2 & 4 & 5 & 2 & 8 & -9 & 3 & 13 \end{cases}$$

$$P_{1} = \begin{cases} 2 & -3 & 3 & 5 & 9 \end{cases}$$

$$P_{1} = \begin{cases} 2 & -3 & 3 & 5 & 9 \end{cases}$$

$$P_{2} = \begin{cases} 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ -3 & 5 & 2 & 8 & -9 & 3 & 13 \end{cases}$$

$$sum(A[R:N]) = pf[R] - pf[R-1] + l70$$

$$sum(A[O:R]) = pf[R] \quad l=0 \text{ is cdgc con}$$

sum (A[R: L])

$$pf[x] = sum(A[0:x])$$

$$pf[x] = sum(A[0:l-1]) + sum(A[1:x])$$

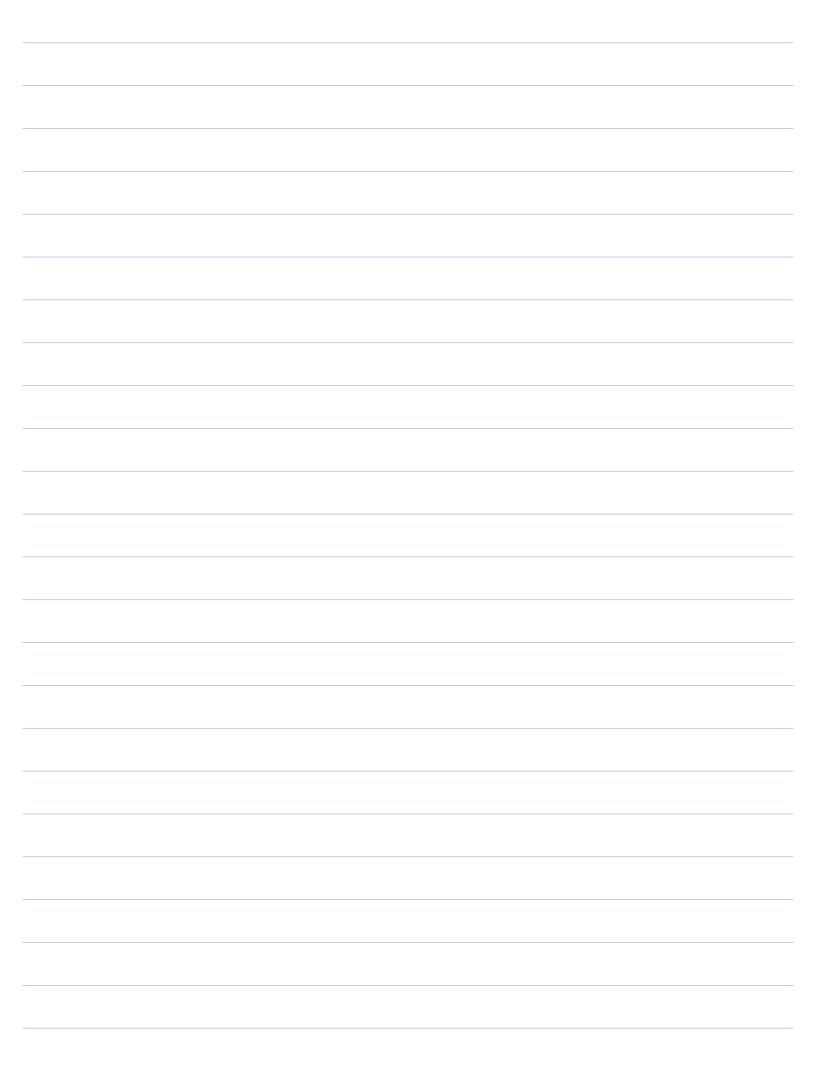
$$pf[x] = pf[l-1] + sum(A[1:x])$$

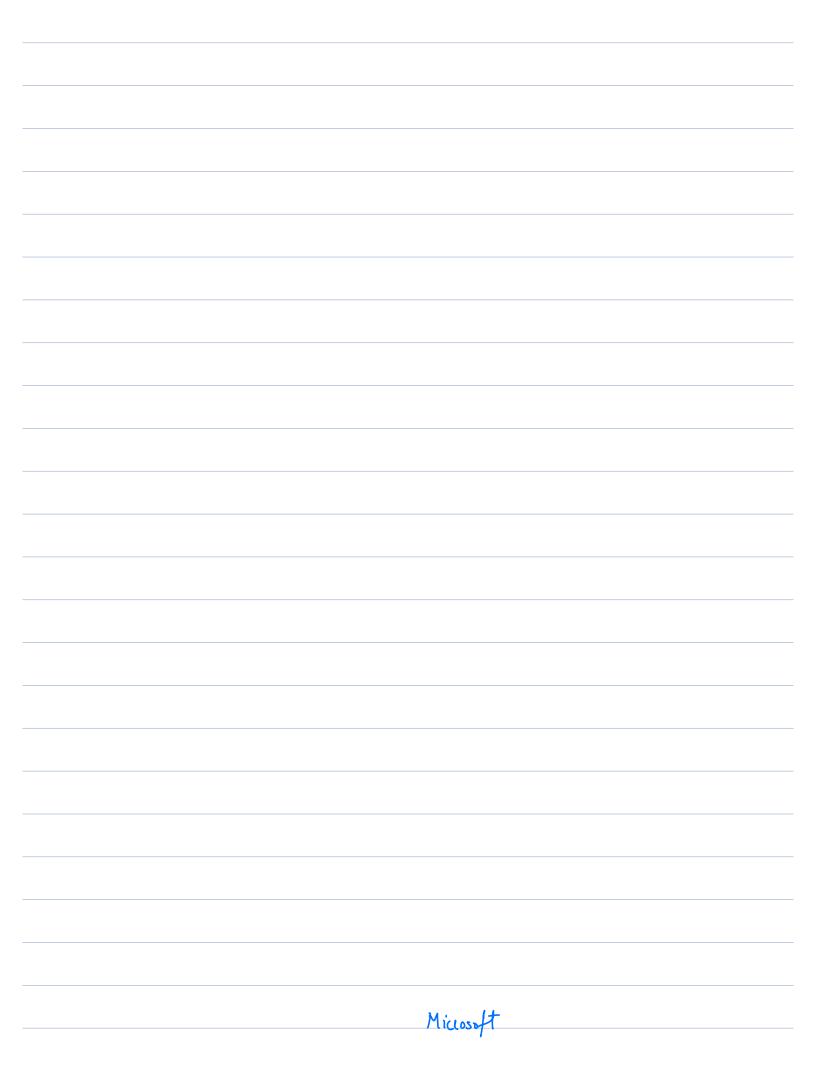
Sum (A[C:N]) =
$$pf[N] - pf[C-1] + l_{70}$$

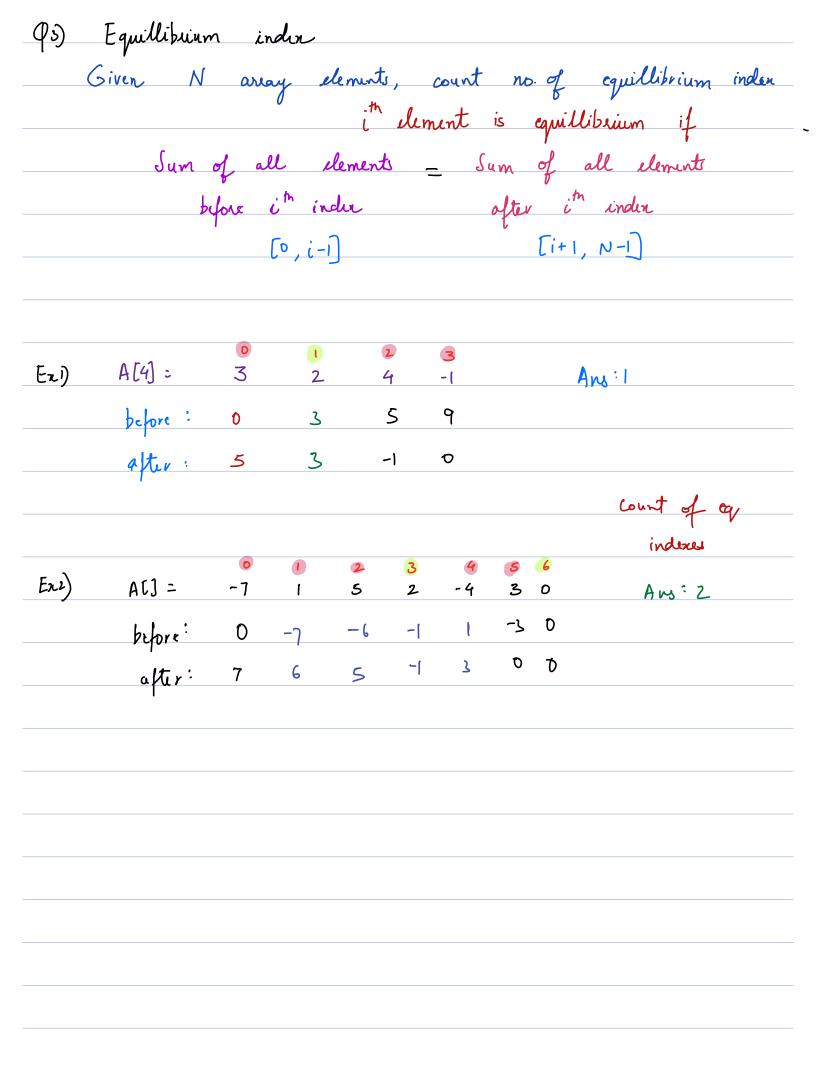
Sum (A[C:N]) = $pf[N]$ l=0 is edge cone
2) for (i=0; i < len(L); i++) \mathcal{E}
 $SI = L[i]$

$$LI = R[i]$$
A[si:ei]
$$A[si:ei]$$

print(sum)







```
count = 0
for (i=0; i<N: i++) {
     before = sum (A[o:i-1])
                                   2 for loops
     after = sum (Alif1: N-13)
                                        TC: 0(N2)
      if (before = = after) {
             count ++
 seturn Count
   sum (A[R: N]) = pf[r] - pf[R-1] + 170
           l= i+1 e= N-1
                                TC: O(N)
Find of array
                                Sc: O(N)
for (i=0; i<N: i++) {
      if (i==0) & before = 03 else & before = pf[i-1]3
      after = sam (A[i+1:N-1]) > pf(N-1) - pf[i]
     if (before = = after) {
             count ++
 seturn Count
                       At i=0 we have an edge
```

