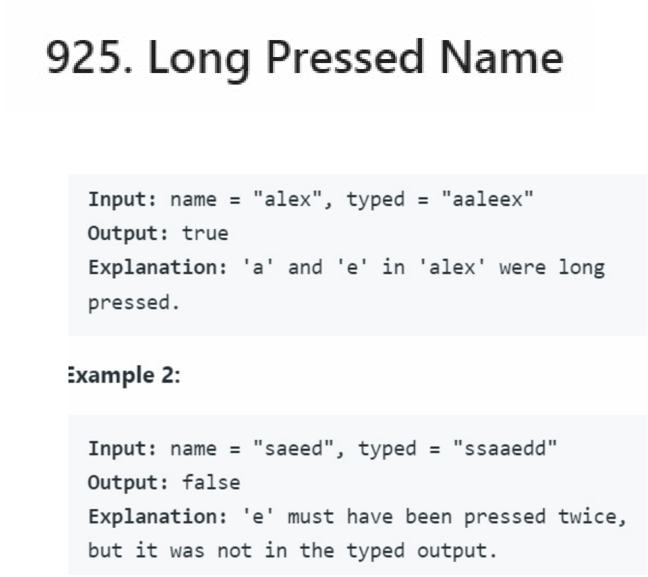


925. Long Pressed Name

```
Input: name = "alex", typed = "aaleex"
Output: true
Explanation: 'a' and 'e' in 'alex' were long
pressed.
```

Example 2:

```
Input: name = "saeed", typed = "ssaaedd"
Output: false
Explanation: 'e' must have been pressed twice,
but it was not in the typed output.
```

name : s a e e d

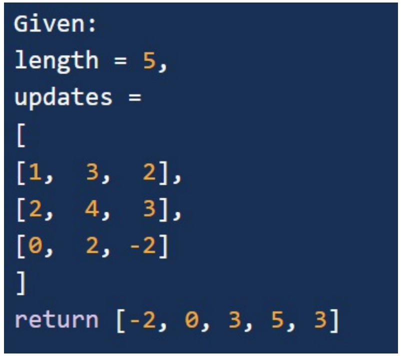
i

typed : s s a a e d

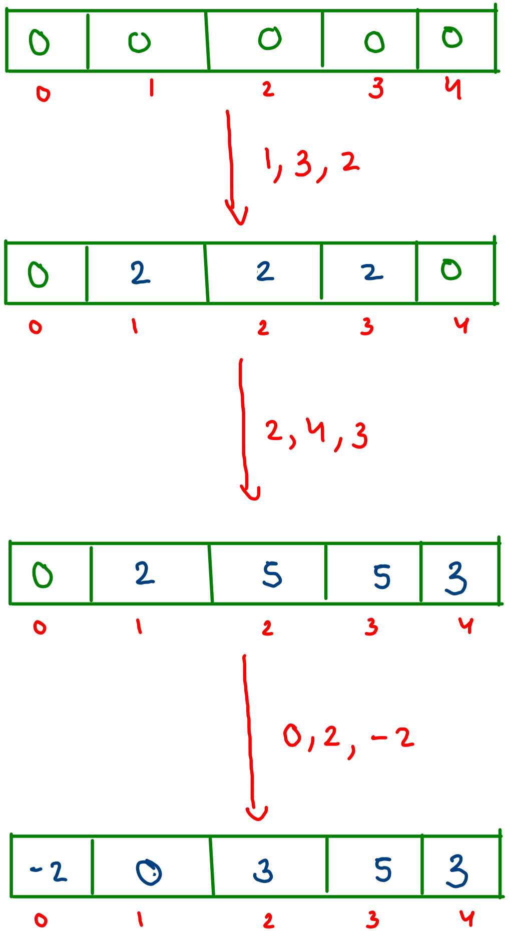
j

```
if ( name[i] == typed [j] ){
        i++ , j++ ;
}
else if ( typed [j] == typed [j-1] ){
        j++;
}
```

903 · Range Addition ✓



```
Given:
length = 5,
updates =
[
[1,  3,  2],
[2,  4,  3],
[0,  2, -2]
]
return [-2, 0, 3, 5, 3]
```

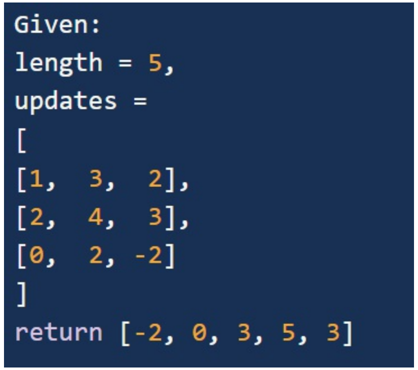


update[i] ->

si, ei, inc

```
for( k : updates.length) {
    int si = updates[k][0];
    int ei = updates[k][1];
    int inc = updates[k][2];
kn
    for(int i = si; i<= ei; i++) {
        arr[i] += inc;
    }
}
```

0	0	0	0	0
0	1	2	3	4



```
Given:
length = 5,
updates =
[
[1,  3,  2],
[2,  4,  3],
[0,  2, -2]
]
return [-2, 0, 3, 5, 3]
```

1 to 3 -> +2

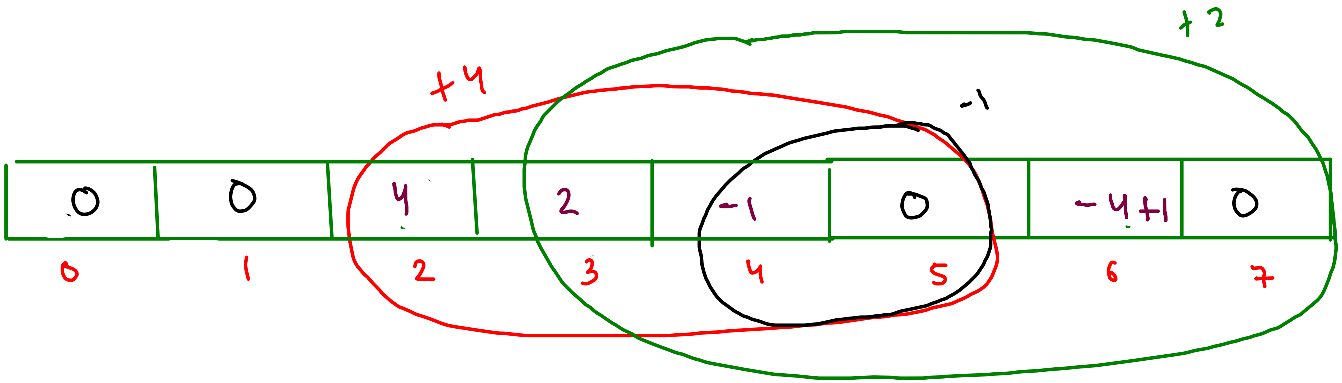
2 to 4 -> +3

0 to 2 -> -2

si, ei, inc

arr (si) = inc

arr (ei +1) = -inc

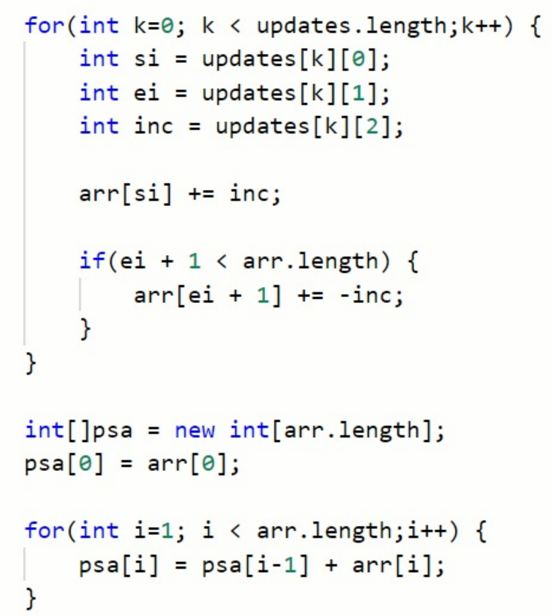


2 to 5 -> +4

3 to 7 -> +2

4 to 5 -> -1

psa 0 0 4 4+2 4+2-1 4+2-1 2 2



```
for(int k=0; k < updates.length;k++) {
    int si = updates[k][0];
    int ei = updates[k][1];
    int inc = updates[k][2];

    arr[si] += inc;

    if(ei + 1 < arr.length) {
        arr[ei + 1] += -inc;
    }
}

int[]psa = new int[arr.length];
psa[0] = arr[0];

for(int i=1; i < arr.length;i++) {
    psa[i] = psa[i-1] + arr[i];
}
```

Max Range Queries

N queries → L to R

1-5, 2-7, 0-3, 6-7

K = 2

1	2	3	3	2	2	2	2
0	1	2	3	4	5	6	7

array after performing all queries

K's count

1	1	2	2	1	1	2	2
0	1	2	3	4	5	6	7

1 - 5 X 4

1	1	2	2	1	1	1	1
0	1	2	3	4	5	6	7

2 - 7 X 2

0	1	2	2	2	2	2	2
0	1	2	3	4	5	6	7

0 - 3 X 6

1	2	3	3	2	2	1	1
0	1	2	3	4	5	6	7

6 - 7 X 3

main array

(i) range addition

(ii)

```
for ( q : queries ) {

        for( int i = L ; i <= R; i--) {
            arr[i] -= 1;
        }
        for ( int i = 0 ; i< arr.length ; i++ ){
            count of K;
        }
        for( int i = L ; i <= R; i--) {
            arr[i] += 1;
        }
}
```

Q * length

1- 5, 2 - 7,
0 - 3, 6- 7

K = 2

1	2	3	3	2	2	2	2
0	1	2	3	4	5	6	7

K K's count till i

0	1	1	1	2	3	4	5
0	1	2	3	4	5	6	7

KP1 (K+1)'s count till i

0	0	1	2	2	2	2	2
0	1	2	3	4	5	6	7

si, ei

after removal of query si to ei, how many K's ?

K's = total - (si to ei K's) + (si to ei (K+1)'s)

= total - (K[ei] - K[si-1]) + (KP1[ei] - KP1[si-1])