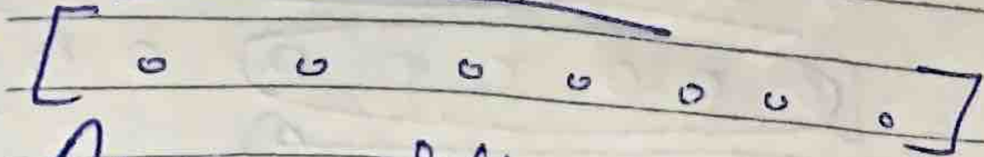
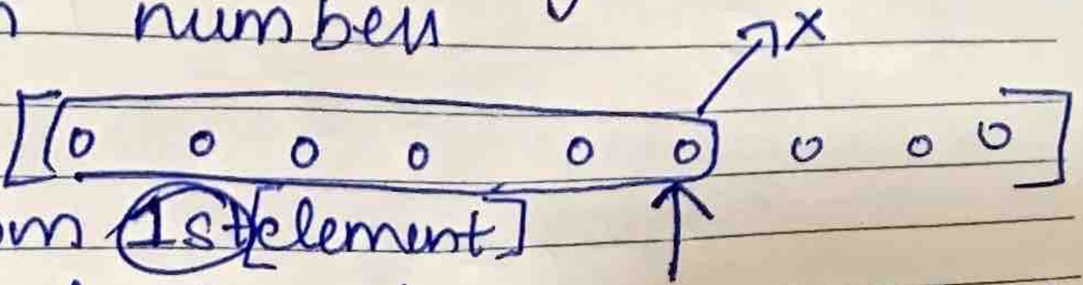
Hashing METHOD



Array with randon no.s

sum = k

Basically take an array with random number



Imagine from [1st element] to the arrowed element, the sum is x taking into account that in that array

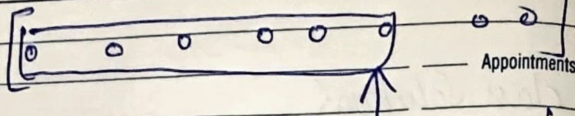
Take this element

there are ~~[illegible]~~ elements 6 elements

March

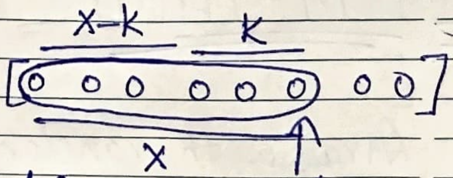
M	T	W	T	F	S	S	M	T	W	T	F	S	S
						1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31					

Appointments



Examining that array again.

The sum of the total elements is x, the required sum is '3' so the remaining sum would be $\boxed{x-k}$



Using this x-k (also known as prefix sum) we would be able to determine if the foward array has a sum of k.

February

M	T	W	T	F	S	S	M	T	W	T	F	S	S
						1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28								

Appointments

For Example

$$arr = \begin{array}{cccccccccc}[1, & 2, & 3, & 1, & 1, & \boxed{1}, & 1, & 4, & 2, & 3] \\ 0 & 1 & 2 & 3 & 4 & \boxed{5} & 6 & 7 & 8 & 9.\end{array}$$

k=3

sum=0 (initially)   i=0.

~~lon~~ctr/long=0.   sum=sum+a[i]

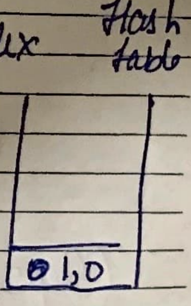
Step 1

sum=0+1 @ 0 index

sum-3
Not app

$(\overset{sum}{x}-i)$

$1-0=1$



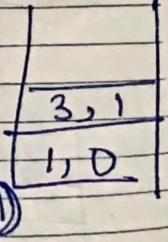
Step 2.

sum=3 @ 1 index

~~aa~~ $\boxed{long=2}$ if (sum==k) i.e.

long = math.max (long, i+1)

↓ it is i+1 since initially starting from 0.



March

M	T	W	T	F	S	S	M	T	W	T	F	S	S
						1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31					

Step3

sum = 6 @ index = 3

sum − k = 3

6 − 3 = 3 contains in hash map

But (2>1) ∴ long = 2 .

6, 2
3, 1
1, 0

Step4

Sum = 7

sum − k = 4 which doesnt contain in hash hence just update the hashmap

7, 3
6, 2
3, 1
1, 0

Step5

sum = 8

sum − k = 5 which doesn contain in (a hash map) hence just update in map

8, 4
7, 3
6, 2
3, 1
1, 0

February

M	T	W	T	F	S	S	M	T	W	T	F	S	S
						1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28								

Step6 sum = 9.

sum − k = 6 } Exists in map.

8, 4
7, 3
6, 2
3, 1
1, 0

long = 2 till now

long = max (2, i − map.get (sum−k)

[5] − 2

⇒ 3

Getting the val indx of sum 6.

And so on - - - -

Hence the code would be

```
fun() (arr, long k)
    ~~long~~ sum = 0;
    ~~an~~ long = 0;
for(int i = 0; i < n; i++)
  { sum = sum + arr[i];
     if sum == k.
  {  ~~ans~~ long = ~~map.get(~~
                 max(long, i+1);
    if ( map.contains (sum−k)
       { long = max (long, i − map.get( sum−k
    if ( ! map. contains)
    map. Put (sum, i) ;
```

March

M	T	W	T	F	S	S	M	T	W	T	F	S	S
						1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31					