Ques) Given n array elements, check if there exists a pair (i, j) s.t.
arr[i] + arr[j] == k && (i != j)

	0	1	2	3	4	5	6	7	8	9
arr[] =	8	9	1	-2	4	5	11	-6	7	5

	i	j
k = 11,	4	8
k = 6,	2	9
k = 22,	6	6	*

$a + b = k$.

1) Create hashset
2) Insert everything in hs.

H.S { 8, 9, 1, -2, 4, 5, 11, -6, 7 }

idea 1 :-
check all pairs.
T.C → $O(n^2)$ S.C → $O(1)$.

```
for (i=0; i<n; i++) {
    a = arr[i]
    b = k - a
    for (j=i+1; j<n; j++) {
        if (arr[j] == b) {
            return True
        }
    }
}
return false
```

k = 11	a	b	in hs.
	8	3	✗
	9	2	✗
	1	10	✗
	-2	13	✗
	4	7	✓

	0	1	2	3	4	5	6	7	8	9
arr[] =	8	9	1	-2	4	5	11	-6	7	5

H.S { 8, 9, 1, -2,
4, 5, 11, -6, 7 }

$\underline{k = 22}$.

a	b	
8	14	✗
9	13	✗
1	21	✗
-2	24	✗
4	18	✗
5	17	✗
11	11	✓

Obs:- If we know freq. we can solve problem.

1) HashMap < int, int > hm.
2) Iterate & insert in hm

hm { 8:1 4:1 7:1
9:1 5:2 -6:1
1:1 11:1 -2:1 }

	0	1	2	3	4	5	6	7	8	9
arr[] =	8	9	1	-2	4	5	11	-6	7	5

k = 22,

a	b	
8	14	*
9	13	*
1	21	*
-2	24	*
4	18	*
5	17	*
11	11	*

⋮

Ex:- 7 5 2 5 9 5 k = 10.

Hm { 7 : 1 9 : 1
5 : 3 2 : 1 }

a	b	
7	3	*
5	5	✓

Pseudo Code :-

1) Create hashMap and insert arr[] 3 O(N)

```
for (i=0; i<N; i++) {
        a= arr[i]
        b= K-a
        if (a!=b) {
            if (b is in map) {          (hm.containsKey(b))
                return True
        }   }
        else {       // a==b
            if (hm[b]>1) {
                return True
          }   }
}
return false
```

T.C → O(N)

T.C → O(N)

S.C → O(N).

* Approach :- using HashSet

	0	1	2	3	4	5	6	7	8	9
arr[] :	8	9	1	-2	4	5	11	-6	7	5

hs

k=22

a	b	hs
8	14	{ }
9	13	{ 8 }
1	21	{ 8 , 9 }
-2	24	{ 8, 9, 1 }
4	18	{ 8, 9, 1, -2 }
5	17	{ 8, 9, 1, -2, 4 }
11	11	{ 8, 9, 1, -2, 4, 5 }
	.	{ 8, 9, 1, -2, 4, 5, 11 }
	.	
	.	

Pseudo code :-

```
hashset <int> hs
for (i=0; i<n; i++) {
        a = arr[i]
        b = k - a;
        if (b is in hs) (
                return True
        }
        else {
                hs.insert (a)
        }
}
return false
```

T.C -> O(n)
S.C -> O(n).

Ques) Calculate no. of i, j s.that,

arr[i] + arr[j] = k.

$i != j$

5 5 5 5

$k = 10$.

↳ Todo :-

Ques) check if there exists a pair s.that, $s[i] - s[j] = k$ $i != j$.

2 3 5 7 9 11 , $k = 2$.

a, b = k + a.

Todo.

10:02 → 10:07

Ques) Given N array elements, calculate no. of distinct elements in every subarray of size k.

k = 4

N = 10, k = 4   [6 9]

	0	1	2	3	4	5	6	7	8	9
arr[] :	2	4	3	8	3	9	4	9	4	10

[0	3]	4
[1	4]	3
[2	5]	3
[3	6]	4
[4	7]	3
[5	8]	2
[6	9]	3

Idea :-
For every window, get no. of distinct using hashset.

Last subarray [N-k, N-1]

$b - a + 1$

$\cancel{N} - \cancel{1} - \cancel{N} + k + \cancel{1} \Rightarrow k$.

[0, n-k],
a b

```
for (i=0; i<= (N-k); i++) {
        Hashset < int > hs;
        for(J = i ; J < (i+k); J++) {
            hs.insert (arr[J])
        }
}
```

T.C → O(N - K+1) * K

S.C → O(K).

$$\begin{array}{lcccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \text{arr} : & 2 & 4 & 3 & \underline{8} & \underline{3} & \underline{9} & \underline{4} & 9 & 4 & 10 \end{array}$$

<u>k=4</u>

<u>Initially</u> :-

	remove	add	HS	Ans
[0-3]			{ ~~2~~, 4, 3, 8 }	4
[1-4]	0	4	{ ~~4~~, 3, 8 }	3
[2-5]	1	5	{ ~~3~~, 8, 9 }	3
[3-6]	2	6	{ 8, 9, 4 }	3

Ans → <u>4</u>

<u>issue</u> :- If we remove an element, indirectly, all occurrence of same element got removed.

11 HashMap with Sliding window :-

	0	1	2	3	4	5	6	7	8	9
arr [] :	2	4	3	8	3	9	4	9	4	10

Subarray :-	remove	add	HashMap	
[0 - 3]			{2:1, 4:1, 8:1, 3:1}	4
[1 - 4]	0	4	{~~2~~:0, 4:1, 8:1, 3:2}	3
[2 - 5]	1	5	{~~4~~:0, 8:1, 3:2, 9:1}	3
[3 - 6]	2	6	{8:1, 3:~~2~~ 1, 9:1, 4:1}	4
[4 - 7]	3	7	{~~8~~:0, 3:1, 9:2, 4:1}	3
[5 - 8]	4	8	{~~3~~:0, 9:2, 4:2}	2
[6 - 9]	5	9	{9:1, 4:2, 10:1}	3.

In hashMap, for our above problem, if freq==0, remove from map.

```
HashMap<Integer, Integer> hm =
            new HashMap<>();
```

Pseudo code :-

```
Hashmap <int , int> hm = new Hashmap<>();
for (i=0; i<k; i++){
    if (hm.containskey (arr[i])){
        hm.put (arr[i], hm.get (arr[i]) +1);
    }
    else {
        hm.put (arr[i], 1);
    }
}
```

K=5

0 1 2 3 4 5 6

```
Print (hm.size());

for (i=1, j=k; i<=(N-k); i++, j++)
    hm.put (arr[i-1], hm.get (arr[i-1]) -1)
    if (hm.get (arr[i-1] ==0) {
        hm.remove (arr[i-1])
    }
    if ( hm.containskey (arr[j]) {
        hm.put (arr[j], hm.get (arr[j]+1)
    } else {
        hm.put (arr[j], 1)
    }
    print ( hm.size() );
}
```

T.C → O(N)

S.C → O(k) .

Doubt

```
Hashmap<int, int> hm;
for (i=0; i<n; i++){
    if (arr[i] is in hm){
        hm[arr[i]]++
```

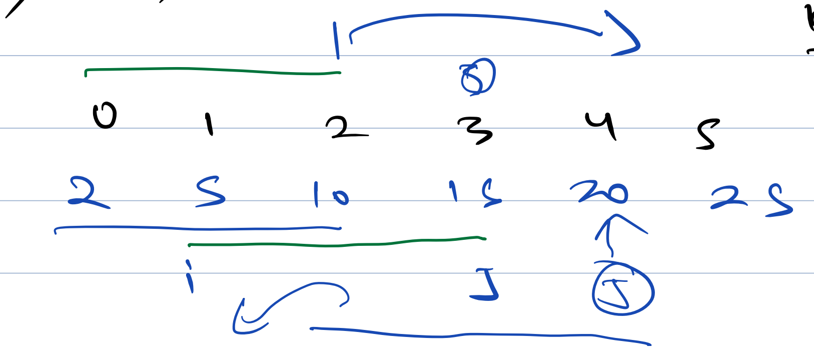
int freq = hm.get(arr[i])

hm.update(arr[i], freq+1)

hm.put(arr[i], hm.get(arr[i])+1)

~~Kreet~~ : ~~Kreet~~ :

k = 3



i + x

$2 \to 1$

$5 \to 1$

$10 \to 1$

$2 \to \cancel{3}\ \cancel{2}\ 1$

~~$3 \to \cancel{2}\ \cancel{1}\ 0$~~

~~$5 \to \cancel{1}\ 0$~~ $\to x$

$6 \to 1$

2, 2, 2, 3, 3, 5, 6

2, 2, 3, 3, 3, 4, 5, 6, 5

2

2 2 3 3 5

2, 2, 3, 3, 5, 6