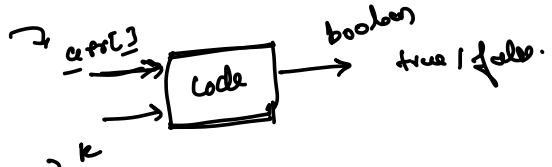
→ Q → Given an array A[] and int k. find if k is present in the array. if it present return true. else return false.

A = [2 , 4 , 9 , 6 , 7 , 11]

k = 11 → true. k = 12 → false.

k = 9 → true k = 0 → false.




```
static boolean search(int[] arr, int k){
    int n = arr.length;
    for(int i=0; i<n; i++){
        if(arr[i] == k){
            return true;
        }
    }
    return false;
}
```

k = 8 k = 20

2, 4, 6, 8

i	k
0	2 == k

k = 6

arr = [2, 4, 6, 8]

return true

① finding element

i	i < n	arr[i] == k	i++
0	T	2 == 6 → F	1
1	T	4 == 6 → F	2
2	T	6 == 6 → T	

return true

② Not found an element.

A = [2, 4, 6, 8]

k = 7

i	i < n	arr[i] == k	i++
0	T	2 == 7 → f	1
1	T	4 == 7 → f	2
2	T	6 == 7 → f	3
3	T	8 == 7 → f	4
4	F → come out of loop		

Q2 → Given A[] and k. find the frequency of k in the array.

n = 9

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

k = 2 → frequency = 2

count = 0, 1, 2

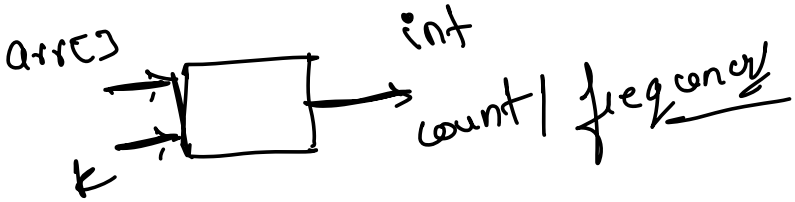
count++

Op 2

k = 3 → 2

k = 9 → 1

k = 5 → 0.



```
static int frequency(int arr[], int k){
    int count = 0;
    for(int i=0; i<arr.length; i++){
        if(arr[i]==k){
            count++;
        }
    }
    return count;
}
```

6

6 < 6 ✗

(i < 6)

k = 2

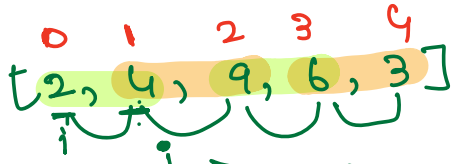
arr = [1, 4, 6, 2, 7, 2]

count	i	arr[i] == k	count++	i++
0	0	1 == 2 → f	—	1
0	1	4 == 2 → f	—	2
0	2	6 == 2 → f	—	3
0	3	2 == 2 → T	1	4
1	4	7 == 2 → f	—	5
1	5	2 == 2 → T	2	6

6 → come out loop.

Olp → 2.

Q3) Given A[] and k. find the difference between adjacent pairs, if the diff is equal k return true else return false.



adjacent pair.
(2,4)
(4,9)
(9,6)
(6,3)

i=0 2,4 → 0, 1 → i+1

i=1 4,9 → 1, 2 → i+1

i=2 9,6 → 2, 3

i	A[i+1] - A[i] => diff	== k == 5.
0	A[1] - A[0] => 4-2 =	(2 == 5) ✗
1	A[2] - A[1] = 9-4 =	(5 == 5) true

done

Solution (wrong soln)

```
static boolean checkdiff(int[] arr, int k){
    int n = arr.length;
    for(int i=0; i<n; i++){
        if(A[i+1] - A[i] == k){
            return true.
        }
    }
    return false;
}
```

(i+1) always handle array out of bound.

k=5, n=4

arr = 2 4 6 3

0 1 2 3

i	i<n	A[i+1] - A[i] == k	i++
0	T	A[1] - A[0] → f (4 - 2)	1
1	T	A[2] - A[1] (6 - 4 = 2 → f)	2
2	T	A[3] - A[2] (8 - 6 = -3 → f)	3
3	T	A[4] - A[3] → index out of bound exception.	

X

Solution correct soln

```
static boolean checkdiff(int[] arr, int k){
    int n = arr.length;
    for(int i=0; i<n-1; i++){
        if(A[i+1] - A[i] == k){
            return true.
        }
    }
    return false;
}
```

(i+1) always handle array out of bound.

k=5, n=4

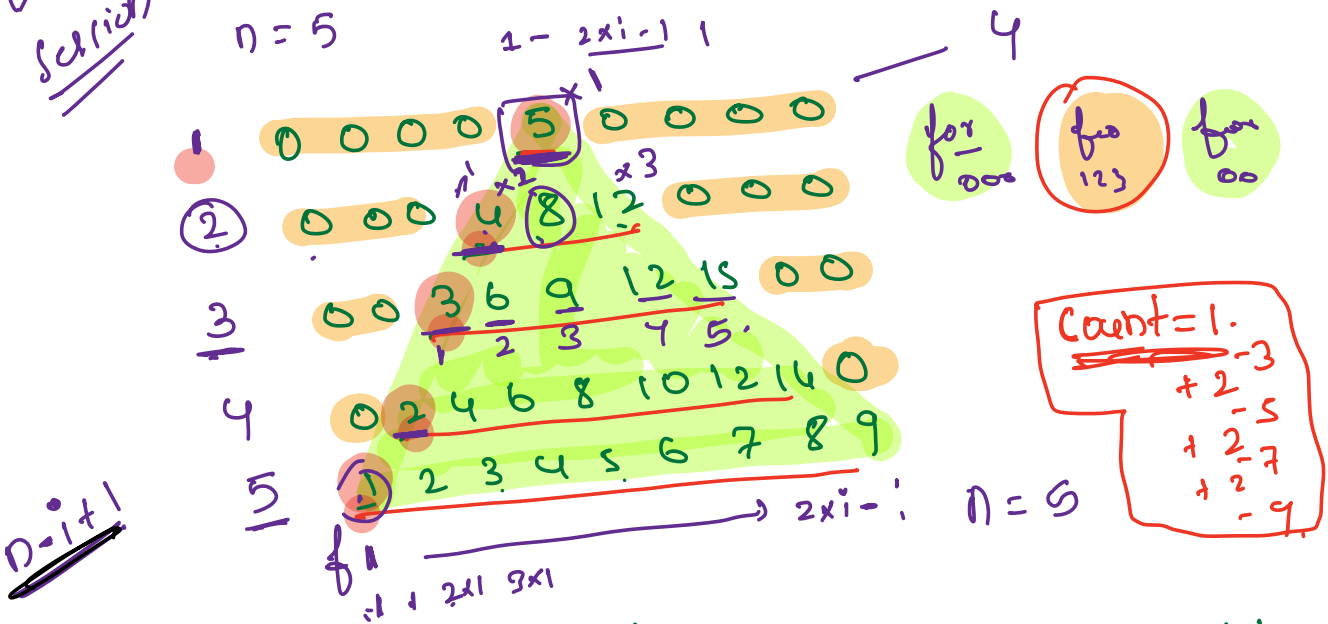
arr = 2 4 6 3

0 1 2 3

i	i<n-1 (loop condition)	A[i+1] - A[i] == k	i++
0	T	A[1] - A[0] → 4 - 2 → F	1
1	T	A[2] - A[1] → 6 - 4 = 2 → F	2
2	T	A[3] - A[2] →	3

3 F → come out of loop.

Pattern



0 0 3 6 9 12 15 0 0

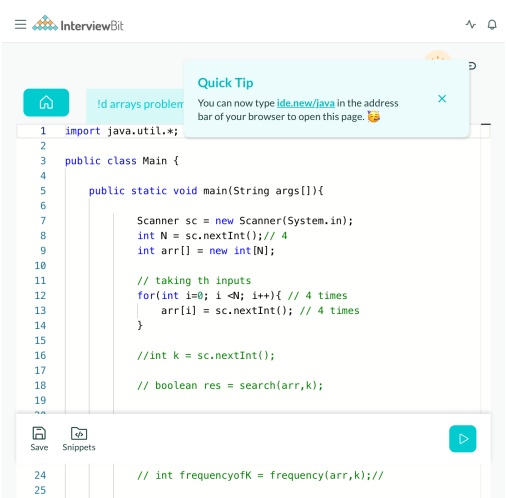
```
int st = n-i+1
for(j=1 ; j<=2i-1 ; j++)
    sop(st+xj)j
```

- 1

i	i/p	count of nos	st. value	right
1	4	1	5	4
2	3	3		3
3	2	5		2
4	1	7		1
5	0	9		0
	(n-i)	(2i-1)		(n+i)

int c = 1; → use for count logic only

```
for(int i=1 ; i <=n ; i++ ) {
    // for first zeros
    for(int j=1 ; j <=n-i ; j++) {
        sop("0");
    }
    int st= n-i+1;
    for(int j=1; j <= C || 2i-1 ; j++){
        sop(st*j);
    }
    C+=2;
    for(int j=1 ; j <=n-i ; j++) {
        sop("0");
    }
    sop(nC);
}
```



https://www.interviewbit.com/snippet/d469fc3b2091d196b93b/