Q) Given an array, return true if A[] is in strictly increasing order, else return false.

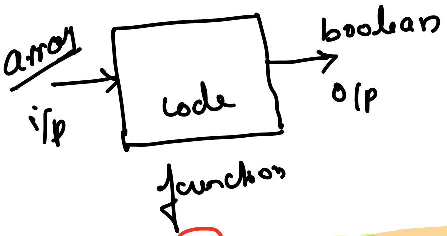
3 5 7 9 12 → true.

3 5 6 9 12 → true.

3 5 10 18 15 → false.

3 5 6 6 12 → false

this is not strictly increasing.



eg:- 3 5 7 9 12

A[i+1] > A[i] ← ↳ compare each pair 1 by 1

A[i+1] < A[i]

A[i], A[i+1]
i next element

3 4 2

3 4 4

4<3 4≤4 true

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

n → A[i+1]
↳ go beyond last element
so stop it (AIOBE)
(n-1)

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

true.

① for correct order

$$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ 3 & 5 & 10 & 11 & 13 \end{array}$$

n = 5

n−1 = 4

i	i<n-1 (4)	A[i+1] ≤ A[i]	return false	i++
0	T	A[1] ≤ A[0] 5 ≤ 3	–	1
1	T	A[2] ≤ A[1] 10 ≤ 5	–	2
2	T	A[3] ≤ A[2] 11 ≤ 10	–	3
3	T	A[4] ≤ A[3] 13 ≤ 11	–	4
4	F → come out of loop. & return true o/p			

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

false

② for wrong order

$$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ 3 & 5 & 10 & 8 & 13 \end{array}$$

n = 5

n−1 = 4

i	i<n-1 (4)	A[i+1] ≤ A[i]	return false	i++
0	T	A[1] ≤ A[0] 5 ≤ 3	–	1
1	T	A[2] ≤ A[1] 10 ≤ 5	–	2
2	T	A[3] ≤ A[2] 8 ≤ 10	return false	3

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

false

false

① for increasing but not strictly

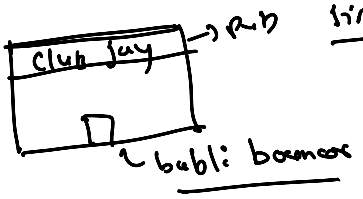
$$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ 3 & 5 & 5 & 5 & 13 \end{array}$$

n = 5

n-1 = 4

i	i<n-1 (4)	A[i+1] ≤ A[i]	return false	i++
0	T	A[1] ≤ A[0] 5 ≤ 3	—	1
1	T	A[2] ≤ A[1] 5 <= 5 T → return false	—	2

Q2) Given an array in which all elements are in pair, coming twice, except one, return the single element if not return (-1.) tve



single,

single person
Stand
+
~~[illegible]~~ out

$3\ 2\ 3\ 9\ 4\ 2\ 9 \rightarrow 4$

$3\ 9\ 6\ 4\ 9\ 3\ 4 \rightarrow 6$

$3\ 4\ 3\ 5\ 4\ 5 \rightarrow -1$

approach /idea.

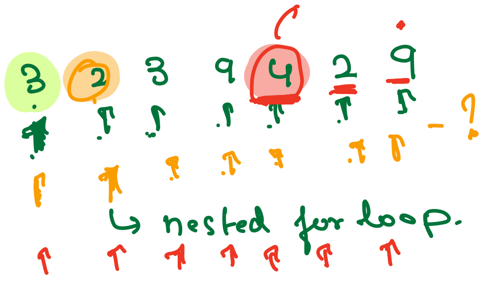
1) Use nested for loop.

2) compare each element with all elements

3) calculate freq

4) if freq==1 return num

5) if no single return -1.



↳ nested for loop.

- ? one loop

box

freq = 0;

3 → 1 → 2

2 → 1 → 2

4 → 1

if(count==1)
return num;

Code

```
for(int i=0; i<n; i++){
        int el = A[i];
        int freq = 0;
        for(int j=0; j<n; j++){
                if(el == A[j]){
                        freq++;
                }
        } //-j loop
        if(freq==1){
                return el;
        }
} i-loop
        return -1;
}
```

Break untill 10:38

PREM CHARAN

Contact for refreshments.

Dry run below

```
for (int i=0; i<n; i++) {
    int el = A[i];
    int freq = 0;
    for (int j=0; j<n; j++) {
        if (el == A[j]) {
            freq++;
        }
    } // i-j loop
    if (freq==1) {
        return el;
    }
} i-loop
return -1;
}
```

eg:

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

→ 1

n= 7

i	el	f	j (j<n)	el ==A[j]	freq++	j++	freq==1	i++
0	3	0	0	el == A[0] 3 = = 3	1	1		
		1	1	el ==A[1] 3 = = 2 → f	–	2		
		0 2	2 ⋮	el == A[2] 3 == 3 → T	2	3	f==1	
					2	7 →	F	1
		2	6	....				
1	2	0		el = A [0 – 6] 2 = [0 — 6] → 2 1, 5		7 →	F	2
2	3	0		el = A[0—6] 3= A[0—6] → 2 (0, 2)		7 ⇒	F	3
3	1	0	0	el == A[0] 1 == 3	–	1		
	1	0	1	el == A[1] 1 == 2	–	2		
	1	0	2	el == A[2] 1 == 3	–	3		
	1	0	3	el == A[3] 1 == 1	1	4		
	1	1	4	el == A[4] 1 ==4	1	5		
	1	1	5	el == A[5] 1 == 2	1 1	6 6		
	1	1	6	el == A[6] 1 == 4	1	6	f==1 1==1	

return el → 1

O/P → 1

Q) Given an array and a value k. Return true if there is pair for which

$A[i] + A[j] == k$

$i != j$

i e_1 e_2 e_3

$\updownarrow$

j e_1 e_2 e_3

(i, j)

else return false.

arr → 3 5 2 1 3 7

0 1 2 3 4 5

k = 5 (0, 2) (2, 4) → true.

k = 2

i, i

$(3, 3)$

$1 + 1 \Rightarrow 2$

$i != j$

(i, j)

pseudo code

take for loop i ←

take inner forloop j

and A[i] and A[j]

compare with k

if == return true

close loops

finally return false.

```
for (i → n)
    for (j → n)
        if (A[i] + [j] == k
                && i != j)
        {
            return true
        }
    }
}
return false.
```

```
for(int i=0; i<n; i++){
    for(int j=0; j<n; j++){
        if(A[i]+A[j]==key i!=j){
            return true;
        }
    }
}
return false;
```

Doubt

→ return type

public int[] solve(int) → return integer array

public int solve( ) ↳ return int

int → int

int[] → Array of integer

$A = [\underline{2}, 3, 4, 5, 6, 2]$

$\underline{A[i]} - \text{min\_ind}(j)$

```
mind_ind(x)
for (i ⟶ N)
    if (A[i] == A[x])
        return i;
```

② 1

j ② 3, 4, 5 0

②

① ① ② ③

+ ↑

② 3 2 3 5

https://www.interviewbit.com/snippet/e1a44c626e076df06f4e/