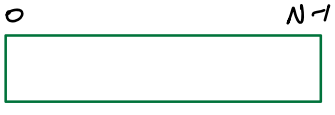
	0	1	2	3	4	5	6
A:	3	5	1	6	2	4	7

A[2] + A[3] ⇒ 1 + 6 ⇒ 7

```
int A[] = new int[6];
SOP(A[5] + A[6]);
```

	0	1	2	3	4	5
A:	0	0	0	0	0	0

Out of bounds error



⇒ N size

Q1: Given an array of size N
Search for element K
Return true if it is present
Else return false

	0	1	2	3	4	5	6
A:	3,	2,	8,	9,	14,	10,	7

K : 8 ⟶ True
K : 11 ⟶ False

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

A: 3, 2, 8, 6
k: 8 ⟶ True

Q: Given an array A
Return the freq (count) of element
k in the array

① ② ③ ④
A: 3, 4, 1, 3, 7, 3, 3, 8
k: 3 ⟶ 4

```
int findCount(int A[], int k) {
    int n = A.length;
```

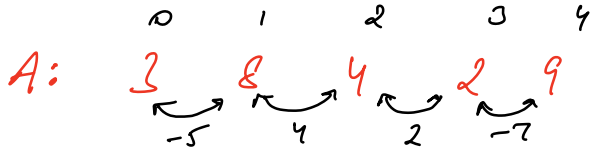
```
int cnt = 0;

for (int i = 0; i < n; i++) {
    if (A[i] == k) {
        cnt++;
    }
}
return cnt;
}
```

Q: Given an array A[]. Return true if diff b/w any adjacent elements is equal to k

adjacent elements ⇓ next to each other

$A[i] - A[i+1] = k$



k: -7 ⇒ true

k: -2 ⇒ false

```
boolean findDiff(int A[], int k) {

    int n = A.length;

    for(int i = 0; i < n-1; i++) {
```

```
if (A[i] - A[i+1] = k){
    return true;
}
}
return false
}
```

	0	1	2	3	4
A:	3	8	4	2	9

Break 10pm

Arraylist

```
int n = 10;
int arr[] = new int[n];
```

Example : Contact, Music library etc

Array ⇒ Static
Arraylist ⇒ Dynamic

```
Arraylist<Integer> arr = new Arraylist<Integer>();
```

⇓

Class

int, float, long, double
×

Integer, Long, Double
Float, String

Basic Operations

Representation

arr:

arr: 2

arr: 2 -1

arr: 2 -1 10
0 1 2

Add

arr.add(2)

arr.add(-1)

arr.add(10)

Get

index

SOP(arr.get(1)); ⇒ -1

SOP(arr.get(0)); ⇒ 2

SOP(arr.get(3)); ⇒ Error

Size

arr.size(); ⇒ 3

Update

arr: 2 -1 10
0 1 2

index new value

arr.set(1, 3);

arr: 2 3 10
0 1 2

arr.set(-1, 4); ⇒ Error