Q.1 Given An array. Write a function that returns both min and max of array.

→

```
int [] minMax (int [] arr) {
    int min = Integer.MAX_VALUE;
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] < min) {
            min = arr[i];
        }
    }
    int max = Integer.MIN_VALUE;
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
    }
    int [] ans = new int [2];
    ans [0] = min;
    ans [1] = max;
    return ans;
}
```

Identity in mathematical function

sum (A, 0) ⟶ A

prod (A, 1) ⟶ A

max (A, −∞) ⟶ A

min (A, ∞) ⟶ A

Q. 2 Given an Array. write a function that construct and return sum array.

	0	1	2	3
A =	1	2	5	3
Sum =	1	3	8	11

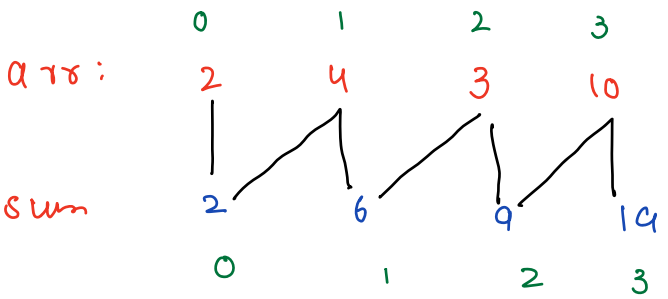
sum[i] represents sum of array values from 0 to i

```
int[] sumArray (int [] arr) {
    int[] sum = new int [arr.length];
    for (int i=0; i< arr.length; i++) {
        // calculate sum from 0 to i
        int temp = 0;
        for (int j=0; j<=i ; j++) {
            temp += arr[j];
        }
        sum[i] = temp;
    }
    return sum;
}
```

```
int[] sumArray(int[] arr) {
    int[] sum = new int[arr.length];
    for (int i=0; i<arr.length; i++) {
        // calculate sum from 0 to i
        int temp = 0;
        for (int j=0; j<=i; j++) {
            temp += arr[j];
        }
        sum[i] = temp;
    }
    return sum;
}
```

arr:	2	4	3
	0	1	2
sum:	2	6	9

i	j	temp
0	0	2
	1 break	
1	0	0+2+
	1	4
	2 break	
2	0	0+2+
	1	4+3
	2	
	3 break	
3		



sum[2] : A[0] + A[1] + A[2]

sum[3] : A[0] + A[1] + A[2] + A[3]

sum[2]

sum[0] = A[0]

sum[1] = sum[0] + A[1]

sum[2] = sum[1] + A[2]

sum[3] = sum[2] + A[3]

sum[i] = sum[i-1] + A[i]

```
int[] sumArray(int[] arr) {
    int[] sum = new int[arr.length];
    sum[0] = A[0];
    for(int i=1; i<arr.length; i++) {
        sum[i] = sum[i-1] + A[i];
    }
    return sum;
}
```

i	
1	sum[1] = sum[0] + A[i]
2	sum[2] = sum[1] + A[2]
3 break	

arr:	2	4	3
	0	1	2
Sum	2	6	9
	0	1	2

Prefix Sum Array

Q.3 Given an Array and value K. write a function that returns true if there exists a pair A[i], A[j] (i != j) such that A[i] + A[j] = K.

target sum pair.

A :

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

K = 10 (1, 4) (2, 3) true

8 2 1 9

K = 7 (0, 6) (4, 5) true

3 4 2 5

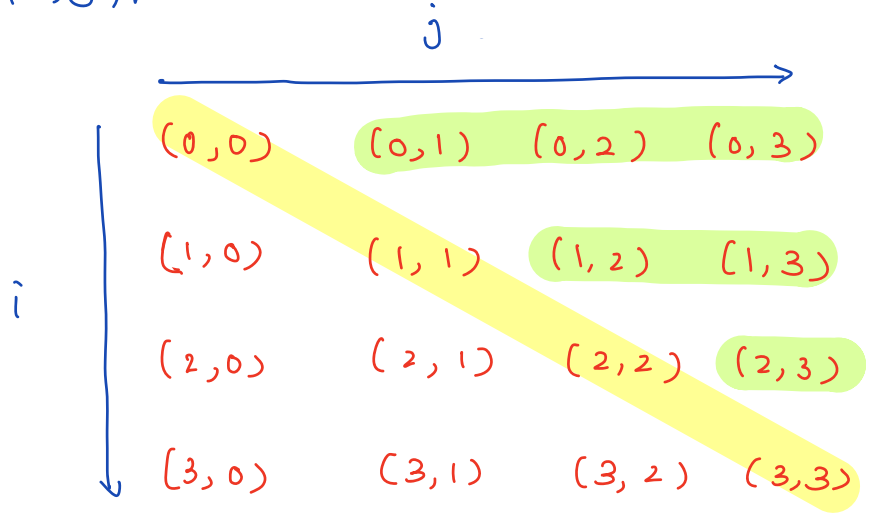
K = 15 false

```
boolean pairSum (int [] arr, int K) {
    for (int i=0; i< arr.length; i++) {
        for (int j=0; j<arr.length; j++) {
            if (i!=j && arr[i] + arr[j] == K) {
                return true;
            }
        }
    }
    return false;
}
```

arr.length = 4

2 3 9 1

all (i,j) pairs

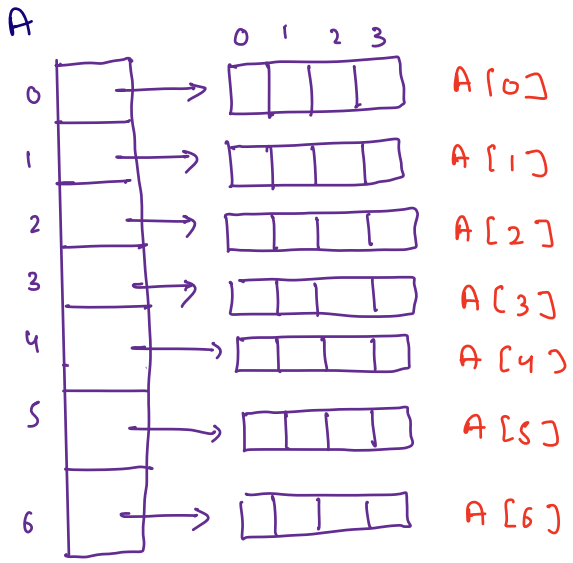


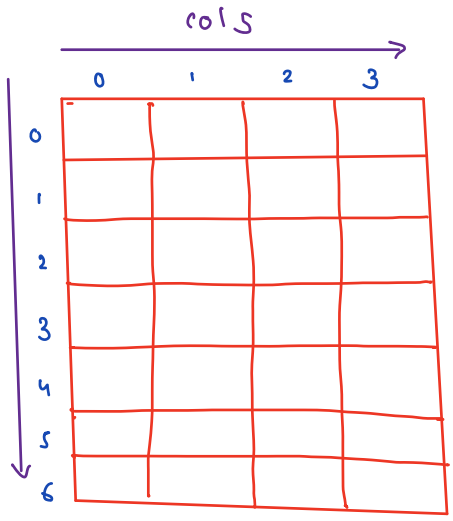
i	j
0	[1-3]
1	[2-3]
2	[3-3]
3	[4-3] nothing

```
for (int i=0; i< arr.length; i++) {
        for (int j =i+1;j <arr.length;j++) {
            if ( arr[i] + arr[j] ==k) {
                return true;
            }
        }
}
return false;
}
```

Improvised approach

2D Arrays





rows = A.length

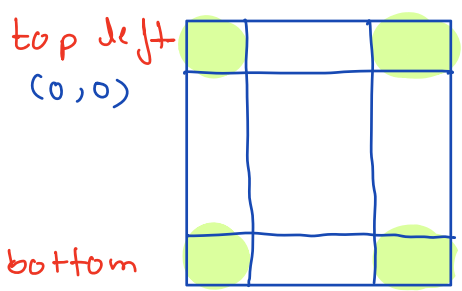
cols = A[0].length

```
int [ ] arr = new int [N];

int [ ] [ ] mat = new int [rows] [cols];
```

	0	1	2
0	0,0	0,1	0,2
1	1,0	1,1	1,2

```
int [ ] [ ] mat = new int [2] [3];
```



rows = N

cols = M

ele = N * M

Q.4 Create a matrix of dimension N x M by taking input from user and print it. (row-wise)

Input : 2 3

10 20 30 40 50 60

output :

	0	1	2
0	10	20	30
1	40	50	60

row-wise print 10 20 30 ↵

40 50 60 ↵

```
void main() {
    Scanner scn = new Scanner();
    int N = scn.nextInt();
    int M = scn.nextInt();
    int[][] mat = new int[N][M];
    for(int i=0; i<N; i++) {
        for(int j=0; j<M; j++) {
            mat[i][j] = scn.nextInt();
        }
    }
    row_wise_print(mat);
}
```

N = 2 M = 3

10 20 30 40 50 60

	0	1	2
0	10	20	30
1	40	50	60

i	j
0	0
	1
	2
	3 break
1	0
	1
	2
	3 break
2	

```
void row_wise_print(int[][] mat) {
    int N = mat.length, M = mat[0].length;
    for(int i=0; i<N; i++) {
        for(int j=0; j<M; j++) {
            System.out.print(mat[i][j] + " ");
        }
        System.out.println();
    }
}
```

3

3

	0	1	2
0	10	20	30
1	40	50	60

10 _ 20 _ 30 _ ↵

40 _ 50 _ 60 _ ↵

N = 2

M = 3

i	j	
0	0	
	1	
	2	
	3	break
1	0	
	1	
	2	
	3	break
2		

```
void col_wise_print(int[][] mat) {
    int N = mat.length, M = mat[0].length;
    for(int j=0; j<M; j++) {
        for(int i=0; i<N; i++) {
            System.out.print(mat[i][j] + " ");
        }
        SOPln();
    }
}
```

	0	1	2
0	10	20	30
1	40	50	60

N = 2

M = 3

10 40 ↓

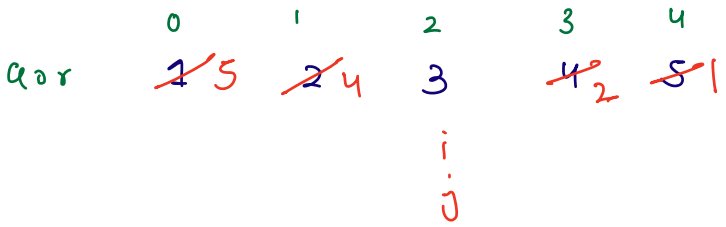
20 50 ↓

30 60 ↓

j	i	
0	0	
	1	
	2	break
1	0	
	1	
	2	break
2	0	
	1	
	2	break

Doubts

Q. reverse an array




```
i = 0,  j = arr.length - 1;
while (i < j) {
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
    i++;
    j--;
}
```