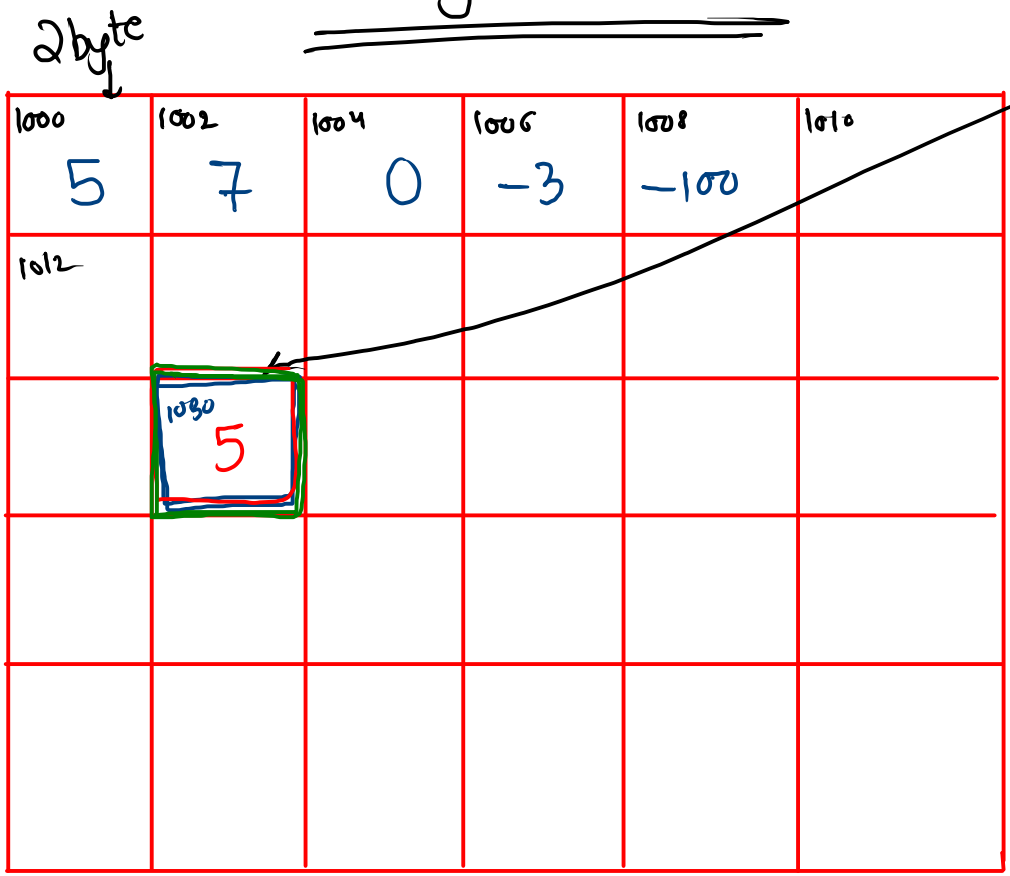
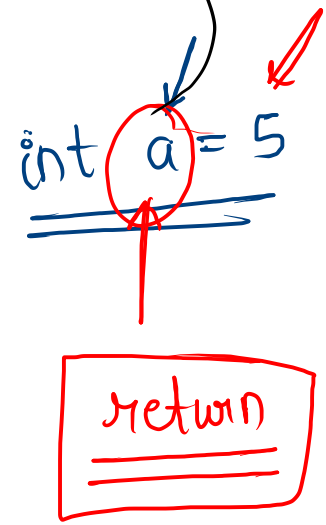
→ Memory allocation



1 Tb



1 Tb

→ 1024 Gb

→ 1024 * 1024 mb

→ 1024 * 1024 * 1024 kb

→ 1024 * 1024 * 1024 * 1 byte

∞ number

code

(array is static means size of array can't be changed)

```
public static void main(String[] args) {
    int[] arr = new int[5];
    // arr_name[index] = value;
    arr[2] = 3;
    arr[1] = 5;
    arr[0] = 4;
    arr[3] = 7;
    arr[2] = 5;
    for (int i = 0; i < 5; i++) {
        System.out.print( arr[i] + " " );
    }
}
```

upgradation

0 0 3 0 0

0 5 3 0 0

4 5 3 0 0

4 5 3 7 0

4 5 5 7 0

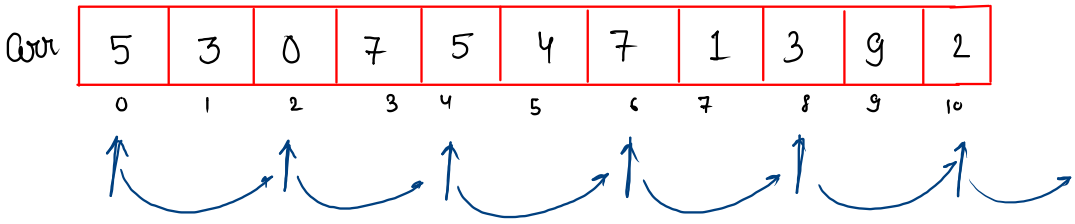
access

Print the array elements linewise

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt(); // 5
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    for (int i = 0; i < n; i++) {
        System.out.println(arr[i]);
    }
}
```

Print Alternate Array Elements Linewise

n = 11



Code

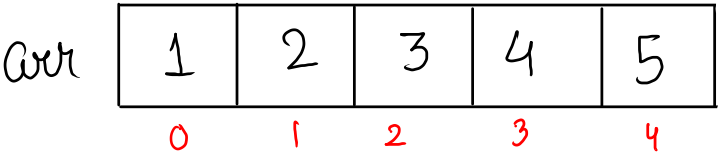
```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    printAlternate(n, arr);
}
public static void printAlternate(int n, int[] arr) {
    for (int i = 0; i < n; i += 2) {
        System.out.println(arr[i]);
    }
}
```

Note:- for loop will only work using indexes of array

Note:- i :- index
arr[i] :- value // arr.length;

Print Array Elements Reverse linewise

n = 5

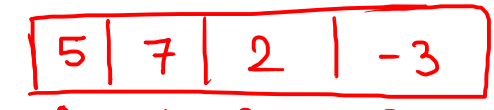


from (n-1) to 0 by -1

code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    printReverse(n, arr);
}
public static void printReverse(int n, int[] arr) {
    for (int i = n - 1; i >= 0; i--) {
        System.out.print(arr[i] + " ");
    }
}
```

n = 4

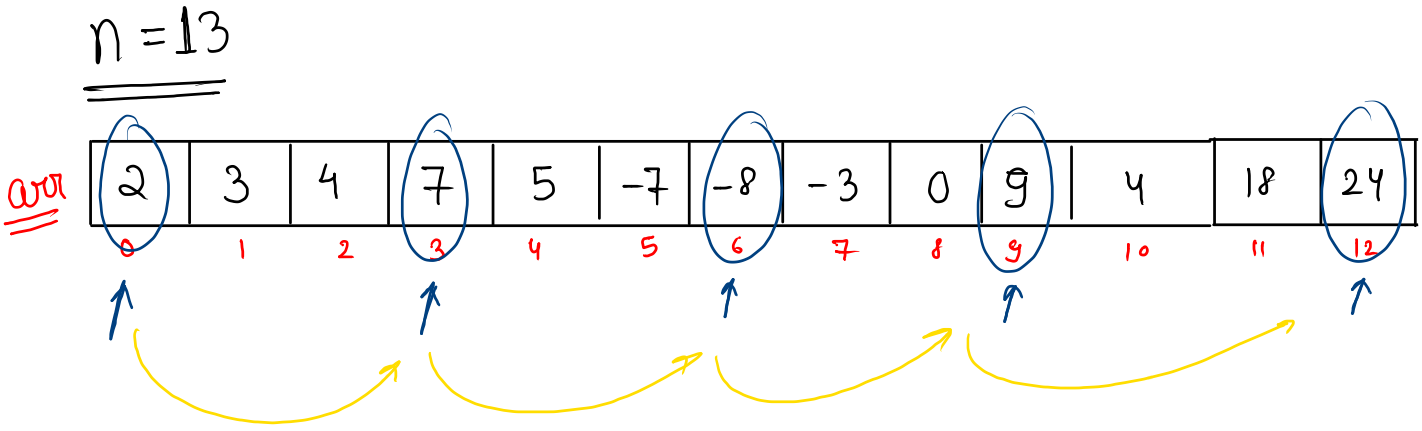


i = 0, arr[0] = 5
i = 1, arr[1] = 7
i = 2, arr[2] = 2
i = 3, arr[3] = -3
i = 4, (4 < 4) ✗

i = 3, (3 >= 0) ✓
i = 2, (2 >= 0) ✓
i = 1, (1 >= 0) ✓
i = 0, (0 >= 0) ✓
i = -1, (-1 >= 0) ✗

-3
2
7
5

Print Array element if index divisible by 3




```
for ( int i = 0; i < n; i++) {
    if ( i % 3 == 0) {
        Syso ( arr [i]);
    }
}
```

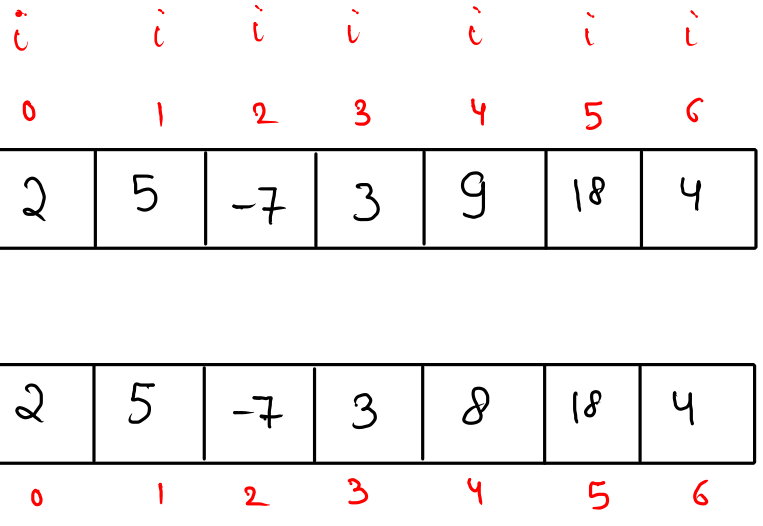
```
for ( int i = 0; i < n; i+=3) {
    Syso ( arr [i]);
}
```

Check if two arrays are identical?

how to check if 2 arrays are same

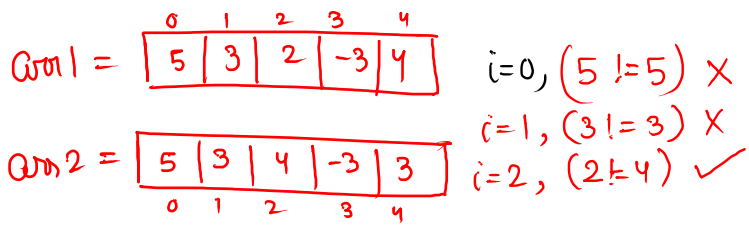
- size of both array's should be same
- value at each index should also be same

Ex:-



code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr1 = new int[n];
    for (int i = 0; i < n; i++) {
        arr1[i] = scn.nextInt();
    }

    int m = scn.nextInt();
    int[] arr2 = new int[m];
    for (int i = 0; i < m; i++) {
        arr2[i] = scn.nextInt();
    }

    System.out.println(checkIdentical(n, arr1, m, arr2));
}

public static boolean checkIdentical(int n, int[] arr1, int m, int[] arr2) {
    if ( n == m ) {

        for (int i = 0; i < n; i++) {
            if ( arr1[i] != arr2[i] ) {
                return false;
            }
        }
        return true;

    } else {
        return false;
    }
}
```



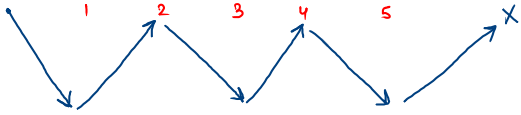
Note:- we will always check opposite cond^n of what question is asking.

Print two arrays alternately

$n = 6$

arr1 =

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



arr2 =

7	8	9	10	11	12
0	1	2	3	4	5

op :- 1 8 3 10 5 12

index

0 → arr1
1 → arr2
2 → arr1
3 → arr2
4 → arr1
5 → arr2