Rotate 7-digit number to right by three

n = 1234567

= 5671234

rem = n % 1000 // 567

n = n / 1000 // 1234

ans = rem * 10000 + n

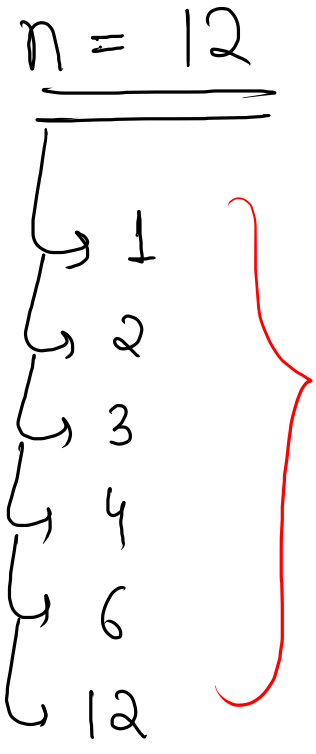
(5670000) + (1234)

(5671234)

code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int t = scn.nextInt();
    for (int i = 0; i < t; i++) {
        int n = scn.nextInt();
        System.out.println(rotate(n));
    }
}
public static int rotate(int n) {
    int rem = n % 1000;
    n = n / 1000;

    return ( rem * 10000 + n );
}
```

Print all factors of a number




```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();

    printFactors(n);
}
public static void printFactors(int n) {

    for (int i = 1; i <= n; i++) {
        if ( n % i == 0 ) {
            System.out.println(i);
        }
    }

}
```

⇒ Arrays (M.M. 9mp)

↳ a collection of similar type of data type

arr (int)

5	3	7	4	0	-5	-100	3
0	1	2	3	4	5	6	7

Syntex

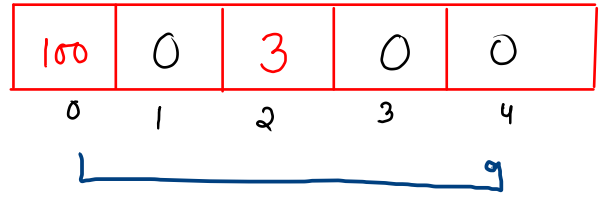
```
data-type [] array_name = new data-type[size];
```

↳ `int[] arr = new int[5];`

0	0	0	0	0
0	1	2	3	4

Note :- In java, default values of array is always going to zero.

```
int [] arr = new int [5];
```




```
arr [2] = 3;   // upgradation
arr [0] = 100;
// arr [5] ⟶ exception : array index out of bound
```

scn.nextInt();

```
int a = arr [4] ;  // 0  // get the value
```

Note :- In String : str.length(); In array : arr.length ; // 5

n = 5

arr

3	4	10	-2	-4
0	1	2	3	4

arr[0] = 3

arr[1] = 4

arr[2] = 10

arr[3] = -2

arr[4] = -4

arr[0];

arr[1];

arr[2];

arr[3];

arr[4];

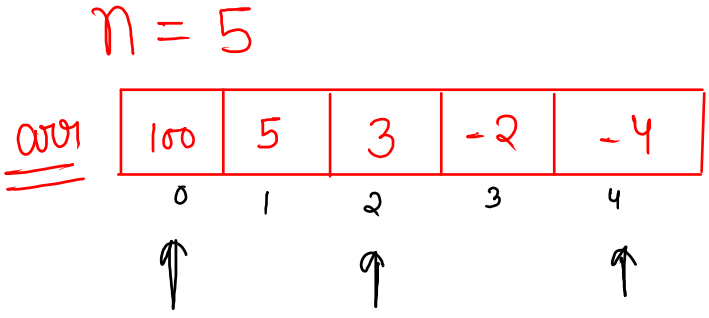
Print the array elements linewise

Code

```
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();
    }
    printArray(arr);
}
public static void printArray(int[] arr) {
    for (int i = 0; i < arr.length; i++) {
        System.out.println( arr[i] );
    }
}
```

Note:- 'i' is index & arr[i] is value

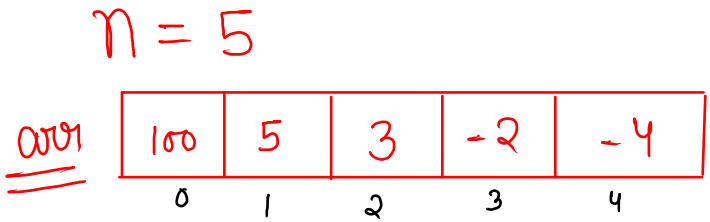
Print Alternate Array Elements Linewise



Code

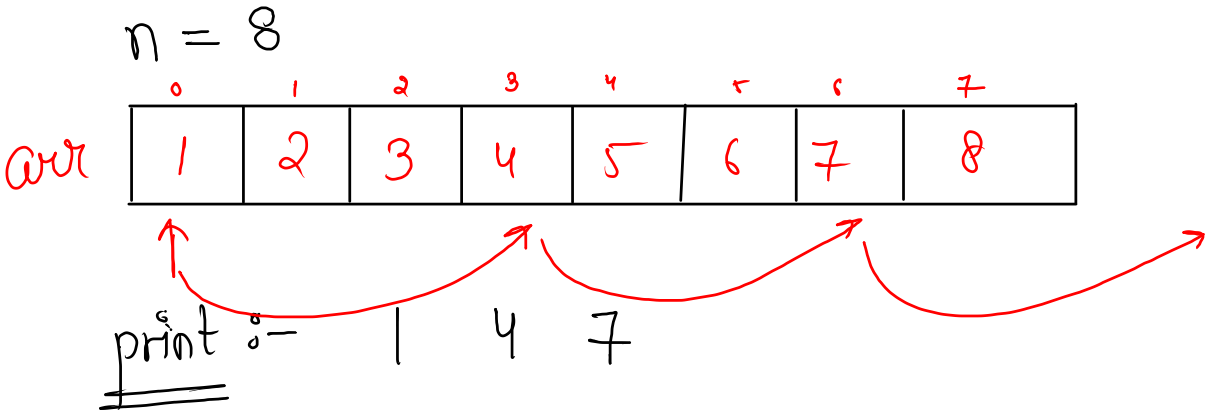
```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    printAlternate(arr, n);
}
public static void printAlternate(int[] arr, int n) {
    for (int i = 0; i < n; i += 2) {
        System.out.println( arr[i] );
    }
}
```

Print Array Elements Reverse linewise



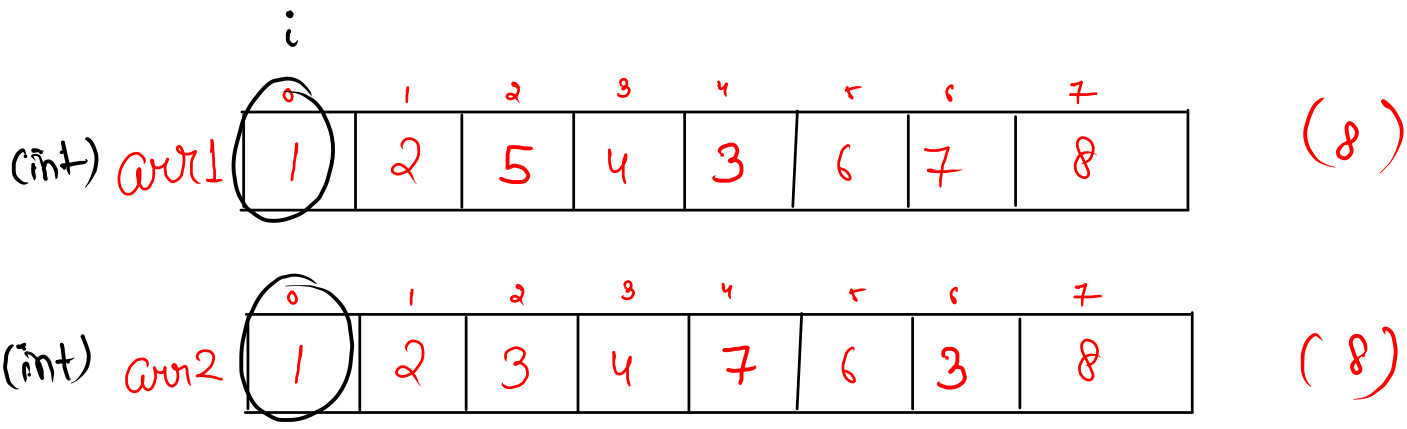

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    printReverse(n, arr);
}
public static void printReverse(int n, int[] arr) {
    for (int i = n - 1; i >= 0; i--) {
        System.out.print(arr[i] + " ");
    }
}
```

Print Array element if index divisible by 3




```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();
    }
    printArray3(n, arr);
}
public static void printArray3(int n, int[] arr) {
    for (int i = 0; i < n; i += 3) {
        System.out.print( arr[i] + " " );
    }
}
```

Check if two arrays are identical?



for 2 arrays to be identical

- both array need to be of same size
- both array should contain same values at same index

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(isIdentical(arr1, arr2));
}
public static boolean isIdentical(int[] arr1, int[] arr2) {

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

}
```