```
2
3 public class ClassA
4 {
5     String[] meth1(int arr1[],char arr2[])
6     {
7         /* Task1 : Pass all the elements which are present in arr1
8                    in a reverse order in another array and print both the arrays
9
10           Task2 : Print all the elements which are present in arr2 by using
11                   for loop (in both directions) & by using for-each loop (One direction)
12
13           Task 3 : You need to implement this task3 in main()
14                    If the length of an array which is returned by meth1() is even
15                    the reterive the first half of the elements.
16
17                    If the length of an array which is returned by meth1() is odd
18                    then reterive only the last two elements of the array
19
20                    Note the length of the array which is returned by meth1()
21                    should be atlest 2.
22          */
23    }
24 }
```

```
3 import java.util.Arrays;
4
5 public class ClassA
6 {
7     String[] meth1(int arr1[],char arr2[]) // arr1=[10,20,30]  arr2=['A','B','C']
8     {
9          //Task1
10        System.out.println("-------Task-1---------");
11
12        int nums[]=new int[arr1.length];
13        int index=0;
14        for(int i=arr1.length-1;i>=0;i--)
15        {
16            nums[index++]=arr1[i];
17        }
18        System.out.println("arr1 : "+Arrays.toString(arr1));
19        System.out.println("nums : "+Arrays.toString(nums));
20
```

```
        //   Task2

        System.out.println("\n-------Task-2---------");
        for(int i=0;i<=arr2.length-1;i++)
            System.out.print(arr2[i]+" ");  // forward
        System.out.println();
        for(int i=arr2.length-1;i>=0;i--)
            System.out.print(arr2[i]+" ");  // backward
        System.out.println();
        for(char c:arr2)
            System.out.print(c+" ");      // forward
```

```
        //| Task 3 :
        return new String[] {"Raju","Ahmed","John"};
    }
    public static void main(String[] args)
    {
        //aobj.meth1(new int[] {10, 20,30},new char[] {'A','B','C'});

        int input1[]= {10,20,30};
        char input2[]= {'A','B','C'};

        ClassA aobj=new ClassA();
        String result[]=aobj.meth1(input1, input2);
        System.out.println("\n-------Task-3--------");
        int len=result.length;
        if(len%2==0)
        {
            for(int i=0;i<len/2;i++)
                System.out.print(result[i]+" ");
        }
        else
        {
            System.out.print(result[len-2]+" "+result[len-1]);
        }
    }
}
```

```
-------Task-1---------
arr1 : [10, 20, 30]
nums : [30, 20, 10]

-------Task-2---------
A B C
C B A
A B C
-------Task-3--------
Ahmed John
```

```
Given an int array length 3, if there is 2 in
the array immediately followed by a 3, change
the element 3 to 0, and return the changed
array.
123---120
234---204
122---122

int[] meth1()
{
    //write your logic here
}
```

```java
package com.pack1;

import java.util.Arrays;

public class ClassA
{
    int[] meth1()
    {
        int arr[]= {4,4,2};

        for(int i=0;i<arr.length-1;i++)
        {
            if(arr[i]==2 && arr[i+1]==3)
                arr[i+1]=0;
        }
        return arr;
    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        int result[]=aobj.meth1();
        System.out.println(Arrays.toString(result
    }
```

```
<terminated> Clas
[4, 4, 2]
```

Given an array of integer, return the sum of the first 2 elements in the array. If the array length is less than 2, just sum up the elements that exist, return 0 if the array length is 0.
1,2,3---3
5,5---10
8,2,1,4---10

```
int meth3(int []arr)
{
   //write your logic here
}
```

```
1  package com.pack1;
2
3  import java.util.Arrays;
4
5  public class ClassA
6  {
7      int meth3(int []arr)
8      {
9          if(arr.length>=2)
10             return arr[0]+arr[1];
11         else if(arr.length==1)
12             return arr[0];
13         else
14             return 0;
15     }
16     public static void main(String[] args)
17     {
18         int input[]= {8,2,1,4};
19         ClassA aobj=new ClassA();
20         int result=aobj.meth3(input);
21         System.out.println(result);
22     }
```

```
<termin
10
```

```
Return an int array length 3 containing the
first 3 digits of pi, {3, 1, 4}.

makePi() ? [3, 1, 4]

public int[] makePi()
{
    // write your logic here
}
```

```java
package com.pack1;

import java.util.Arrays;

public class ClassA
{
    public int[] makePi()
    {
        return new int[] {3,1,4};
    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        int result[]=aobj.makePi();
        System.out.println(Arrays.toString(result));
    }

}
```

```
<terminated> Clas
[3, 1, 4]
```

```
-----------------------------
Given an array of ints length 3, return an
array with the elements "rotated left" so {1,
2, 3} yields {2, 3, 1}.


rotateLeft3([1, 2, 3]) ? [2, 3, 1]
rotateLeft3([5, 11, 9]) ? [11, 9, 5]
rotateLeft3([7, 0, 0]) ? [0, 0, 7]

public int[] rotateLeft3(int[] nums)
{
    // write your logic here
}
```

```java
package com.pack1;

import java.util.Arrays;

public class ClassA
{
    public int[] rotateLeft3(int[] nums)
    {
        return new int[] {nums[1],nums[2],nums[0]};
    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        int input[]= {5, 11, 9};
        int result[]=aobj.rotateLeft3(input);
        System.out.println(Arrays.toString(result));
    }

}
```

```
<terminated> Class
[11, 9, 5]
```

Assignments

Given an int array length 3, if there is 2 in the array immediately followed by a 3, change the element 3 to 0, and return the changed array.
123---120
234---204
122---122

```
int[] meth1()
{
   //write your logic here
}
```

Given an array of integer, return the sum of the first 2 elements in the array. If the array length is less than 2, just sum up the elements that exist, return 0 if the array length is 0.
1,2,3---3
5,5---10
8,2,1,4---10

```
int meth3(int []arr)
```

```
{
  //write your logic here
}
```

Given an array of ints, return true if 6 appears as either the first or last element in the array. The array will be length 1 or more.

firstLast6([1, 2, 6]) ? true
firstLast6([6, 1, 2, 3]) ? true
firstLast6([13, 6, 1, 2, 3]) ? false

```
public boolean firstLast6(int[] nums)
{
 //write your logic here
}
```

Given an array of ints, return true if the array is length 1 or more, and the first element and the last element are equal.

sameFirstLast([1, 2, 3]) ? false

sameFirstLast([1, 2, 3, 1]) ? true
sameFirstLast([1, 2, 1]) ? true

```
public boolean sameFirstLast(int[] nums)
{
   //write your logic here
}
```

Return an int array length 3 containing the first 3 digits of pi, {3, 1, 4}.

makePi() ? [3, 1, 4]

```
public int[] makePi()
{
    // write your logic here
}
```

Given 2 arrays of ints, a and b, return true if they have the same first element or they have the same last element. Both arrays will be length 1 or more.

commonEnd([1, 2, 3], [7, 3]) ? true
commonEnd([1, 2, 3], [7, 3, 2]) ? false

commonEnd([1, 2, 3], [1, 3]) ? true

```
public boolean commonEnd(int[] a, int[] b)
{

   // write your logic here
}
```

Given an array of ints length 3, return the sum of all the elements.

sum3([1, 2, 3]) ? 6
sum3([5, 11, 2]) ? 18
sum3([7, 0, 0]) ? 7

```
public int sum3(int[] nums)
{
   // write your logic here

}
```

Given an array of ints length 3, return an array with the

elements "rotated left" so {1, 2, 3} yields {2, 3, 1}.

rotateLeft3([1, 2, 3]) ? [2, 3, 1]
rotateLeft3([5, 11, 9]) ? [11, 9, 5]
rotateLeft3([7, 0, 0]) ? [0, 0, 7]

```
public int[] rotateLeft3(int[] nums)
{
   // write your logic here
}
```

Given an array of ints length 3, figure out which is larger, the first or last element in the array, and set all the other elements to be that value. Return the changed array.

maxEnd3([1, 2, 3]) ? [3, 3, 3]
maxEnd3([11, 5, 9]) ? [11, 11, 11]
maxEnd3([2, 11, 20]) ? [20,20,20]

```
public int[] maxEnd3(int[] nums)

{
```

```
   // write your logic here
}
```

Given 2 int arrays, a and b, each length 3, return a new array length 2 containing their middle elements.

middleWay([1, 2, 3], [4, 5, 6]) ? [2, 5]
middleWay([7, 7, 7], [3, 8, 0]) ? [7, 8]
middleWay([5, 2, 9], [1, 4, 5]) ? [2, 4]

```
public int[] middleWay(int[] a, int[] b)
{
   // write your logic here
}
```

Given an array of ints, return a new array length 2 containing the first and last elements from the original array. The original array will be length 1 or more.

makeEnds([1, 2, 3]) ? [1, 3]
makeEnds([1, 2, 3, 4]) ? [1, 4]
makeEnds([7, 4, 6, 2]) ? [7, 2]

```
public int[] makeEnds(int[] nums)
{
    // write your logic here
}
```

Given an int array, return a new array with double the length where its last element is the same as the original array, and all the other elements are 0. The original array will be length 1 or more.

makeLast([4, 5, 6]) ? [0, 0, 0, 0, 0, 6]
makeLast([1, 2]) ? [0, 0, 0, 2]
makeLast([3]) ? [0, 3]

```
public int[] makeLast(int[] nums)
{
   // write your logic here
}
```

In java array we have 1-Dimensional array

```java
package com.Array45;
import java.util.Arrays;
public class ClassN //1-d array
{
    void meth1()
    {


    int arr1[]=new int[4];
    //arr1= {22,33,44,55};//invalid
    arr1[0]=1;
    arr1[1]=2;
    arr1[2]=4;

    int arr2[]=new int[] {22,33,55};
    int arr3[]= {22,11,33,44};
    System.out.println(Arrays.toString(arr1));//1st way

    for(int i=0;i<arr1.length;i++)//2nd way
    System.out.print(arr1[i]+" ");

    }
    public static void main(String[] args)
    {
        ClassN nobj=new ClassN();
        nobj.meth1();
    }
}
```

```
<terminated> ClassN (2)
[1, 2, 4, 0]
1 2 4 0
```

To see the array, we use Arrays.toString(arrayname) in 1-d array

2D-Array

To see array in output we need to 2-process

1.Arrays.deeptoString

2.by using for loop

```java
package com.Array45;
import java.util.Arrays;
public class ClassO //2-D array
{
    void meth1()
    {
        int arr1[][];
        arr1=new int[2][3];
        arr1[0][0]=1;
        arr1[0][1]=2;
        arr1[0][2]=3;
        arr1[1][0]=4;
        arr1[1][1]=5;
        arr1[1][2]=6;

        int arr2[][]=new int[][] {{2,3,4},{5,6,7}};

        int arr3[][]= {{9,8,7},{6,5,4}};

        int row=arr1.length;

        int coloumn=arr1[0].length;

        int totalElement=row*coloumn;

        System.out.println("The length of a= "+totalElement);

        System.out.println("aar1 lenth : "+arr1.length);

        System.out.println("aar2 lenth : "+arr2.length);

        System.out.println("aar3 lenth : "+arr3.length);

        System.out.println(Arrays.deepToString(arr1));//1st way



        for(int i=0;i<arr1.length;i++)
        {
            for(int j=0;j<arr1[i].length;j++)
            {
                System.out.print(arr1[i][j]+" ");
            }
            System.out.println();
        }

        System.out.println("-------");
        System.out.println(Arrays.deepToString(arr2));

        System.out.println(Arrays.deepToString(arr3));
        System.out.println(arr1);
        System.out.println(arr1.toString());



    }
    public static void main(String[] args)
    {
        new ClassO().meth1();

    }

}
```

```
<terminated> ClassO (3) [Java Application]
The length of a= 6
aar1 lenth : 2
aar2 lenth : 2
aar3 lenth : 2
[[1, 2, 3], [4, 5, 6]]
1 2 3
4 5 6
-------
[[2, 3, 4], [5, 6, 7]]
[[9, 8, 7], [6, 5, 4]]
[[I@4c203ea1
[[I@4c203ea1
```

3D/multidimensional array

To see array in output we need to 2-process

1.Arrays.deeptoString

2.by using for loop

```java
package com.Array45;

import java.util.Arrays;

public class ClassP //3-D array/multi dimensional array
{
    void meth1()
    {
        int arr2[][][]=new int[2][2][3];
        int arr1[][][]={ {{1,2,3},{4,5,6}},
                { {1,2,3},{4,5,6}}};

        //System.out.println(arr1.length);
        int layer=arr2.length;
        System.out.println("no.of layers "+layer);

        int row=arr2[0].length;

        System.out.println("no.of rows "+row);

        int coloumn=arr2[0][0].length;
        System.out.println("no.of coloumn"+coloumn);

        int totalele=layer*row*coloumn;
        System.out.println("total no of elements "+totalele);

        System.out.println(Arrays.deepToString(arr1));//1st way

```

```
<terminated> ClassP (2) [Java Application] C:\Users\ADMIN\Downloa
no.of layers 2
no.of rows 2
no.of coloumn3
total no of elements 12
[[[1, 2, 3], [4, 5, 6]], [[1, 2, 3],
Layer 0:
1 2 3
4 5 6

Layer 1:
1 2 3
4 5 6
```

```
29          for(int i=0;i<arr1.length;i++)//2nd way
30          {
31          System.out.println("Layer "+i+":");
32          for(int j=0;j<arr1[i].length;j++)
33          {
34          for(int z=0;z<arr1[i][j].length;z++)
35          {
36              System.out.print(arr1[i][j][z]+" ");
37          }
38          System.out.println();
39          }
40          System.out.println();
41          }
42      }
43      public static void main(String[] args)
44      {
45           ClassP pobj=new ClassP();
46           pobj.meth1();
47      }
48
49  }
```