Learning Guide------ARRAYS

Definition

- In computer science, an array is a data structure consisting of a collection of elements.

1. One-dimensional Arrays

(1) Declaration & Initialization

```
int[] a = new int[length];
boolean[] b = new boolean[length];
String[] s = new String[length];
```

Also as

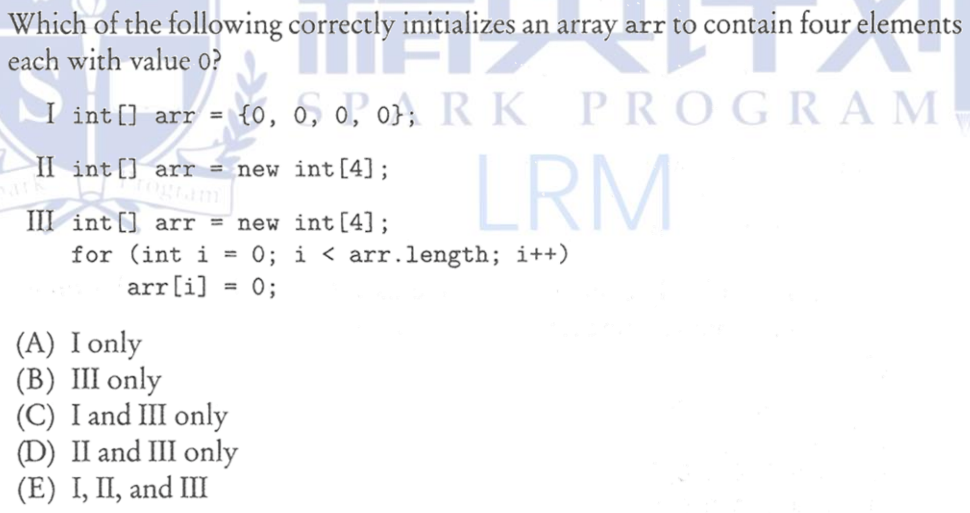
```
int a[] = new int[length];
boolean b[] = new boolean[length];
String s[] = new String[length];                    ------not suggested
```

a[i] refers to the i th element in the array a

Initialize constant array by   int[] a = {1,2,3,4,5};

Initialize the whole array with the same element by

```
for (int i = 0; i < a.length; i++)
        a[i] = 1;
```



Which of the following correctly initializes an array arr to contain four elements each with value 0?

```
 I  int[] arr = {0, 0, 0, 0};

 II int[] arr = new int[4];

III int[] arr = new int[4];
    for (int i = 0; i < arr.length; i++)
        arr[i] = 0;
```

(A) I only
(B) III only
(C) I and III only
(D) II and III only
(E) I, II, and III

(2) Length of Array

```
int length = a.length;
```

Note

1. To return the length of an array, a.length is correct while a.length() is incorrect.
2. Since the array subscripts go from 0 to a.length – 1; therefore, the test on i in the for loop must be strictly less than a.length.

Refer to the following code segment. You may assume that arr is an array of int values.

```
int sum = arr[0], i = 0;
while (i < arr length)
{
    i++;
    sum += arr[i];
}
```

Which of the following will be the result of executing the segment?
(A) Sum of arr[0], arr[1],   , arr[arr.length-1] will be stored in sum.
(B) Sum of arr[1], arr[2],   , arr[arr.length-1] will be stored in sum.
(C) Sum of arr[0], arr[1],   , arr[arr.length] will be stored in sum.
(D) An infinite loop will occur.
(E) A run-time error will occur.

(3) Traversing an Array

for loop or for-each loop can both be used when traversing an array
for loop is used when______and_____elements.
for-each loop is used when______elements.

Refer to the following code segment. You may assume that array arr1 contains elements arr1[0], arr1[1], ..., arr1[N-1], where N = arr1.length.

```
int count = 0;
for (int i = 0; i < N; i++)
    if (arr1[i] != 0)
    {
        arr1[count] = arr1[i];
        count++;
    }
int[] arr2 = new int[count];
for (int i = 0; i < count; i++)
    arr2[i] = arr1[i];
```

If array arr1 initially contains the elements 0, 6, 0, 4, 0, 0, 2 in this order, what will arr2 contain after execution of the code segment?
(A) 6, 4, 2
(B) 0, 0, 0, 0, 6, 4, 2
(C) 6, 4, 2, 4, 0, 0, 2
(D) 0, 6, 0, 4, 0, 0, 2
(E) 6, 4, 2, 0, 0, 0, 0

Consider this program segment:

```
for (int i = 2; i <= k; i++)
    if (arr[i] < someValue)
        System.out.print("SMALL");
```

What is the maximum number of times that `SMALL` can be printed?

(A) 0
(B) 1
(C) k - 1
(D) k - 2
(E) k

(4) Arrays Package

import java.util.Arrays;

Arrays.sort(a); //sort the array a in the increasing order
Arrays.toString(a); //return a String that contains all the elements in array a

Practice

The following code fragment is intended to find the smallest value in `arr[0]   arr[n-1]`

```
/** Precondition:
 *   - arr is an array, arr.length = n.
 *   - arr[0]  .arr[n-1] initialized with integers.
 * Postcondition: min = smallest value in arr[0]...arr[n-1].
 */
int min = arr[0];
int i = 1;
while (i < n)
{
    i++;
    if (arr[i] < min)
        min = arr[i];
}
```

This code is incorrect. For the segment to work as intended, which of the following modifications could be made?

I Change the line

`int i = 1;`

to

`int i = 0;`

Make no other changes.

II Change the body of the `while` loop to

```
{
    if (arr[i] < min)
        min = arr[i];
    i++;
}
```

Make no other changes.

III Change the test for the `while` loop as follows:

```
    while (i <= n)
```

Make no other changes.

(A) I only
(B) II only
(C) III only
(D) I and II only
(E) I, II, and III

2. Two-dimensional Arrays

(1) Declaration & Initialization

```
int[][] table = new int[3][4];
String[][] s = new String[5][5];
boolean[][] boo = new boolean[7][7];
```

Specify a two-dimensional array in this way:

```
int[][] mat = { {3, 4, 5},
{4, 5, 6},
{5, 6, 7},
{6, 7, 8} };
```

Also as `int[][] mat = { {3, 4, 5}, {4, 5, 6}, {5, 6, 7}, {6, 7, 8} };`

a[i][j] refers to the element in the row i and column j

(2) Length of Array

a.length	return the length of rows
a[i].length	return the length of row i

(3) Traversing an Array

for loop

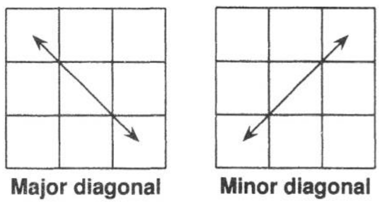
```
for (int i = 0; i < a.length; i++)
    for (int j = 0; j < a[i].length; j++)
        <process>
```

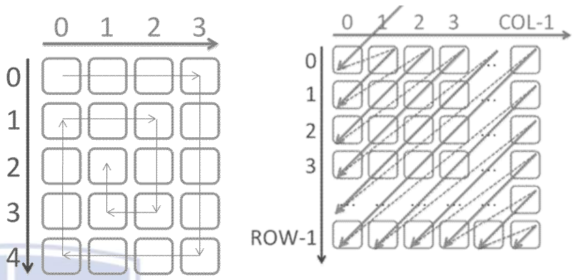
for-each loop
```
for (int[] row : a)
    for (int num : row)
        <process>
```

Think about it!

How to traversing a two-dimensional array in the following ways:

a. major diagonal
b. minor diagonal
c. Back traversal
d. Snake traversal





Practice

Consider a class that has this private instance variable:

```
private int[][] mat;
```

The class has the following method, alter.

```
public void alter(int c)
{
    for (int i = 0; i < mat.length; i++)
        for (int j = c + 1; j < mat[0].length; j++)
            mat[i][j-1] = mat[i][j];
}
```

If a 3 × 4 matrix mat is

```
1 3 5 7
2 4 6 8
3 5 7 9
```

then alter(1) will change mat to

(A)
```
1 5 7 7
2 6 8 8
3 7 9 9
```

(B)
```
1 5 7
2 6 8
3 7 9
```

(C)
```
1 3 5 7
3 5 7 9
```

(D)
```
1 3 5 7
3 5 7 9
3 5 7 9
```

(E)
```
1 7 7 7
2 8 8 8
3 9 9 9
```

The method changeNegs below should replace every occurrence of a negative integer in its matrix parameter with 0.

```
/** @param mat the matrix
 *  Precondition: mat is initialized with integers
 *  Postcondition: All negative values in mat replaced with 0.
 */
public static void changeNegs(int[][] mat)
{
    /* code */
}
```

Which is correct replacement for /* *code* */?

```
I   for (int r = 0; r < mat.length; r++)
        for (int c = 0; c < mat[r].length; c++)
            if (mat[r][c] < 0)
                mat[r][c] = 0;

II  for (int c = 0; c < mat[0].length; c++)
        for (int r = 0; r < mat length; r++)
            if (mat[r][c] < 0)
                mat[r][c] = 0;

III for (int[] row : mat)
        for (int element : row)
            if (element < 0)
                element = 0;
```

(A) I only
(B) II only
(C) III only
(D) I and II only
(E) I, II, and III