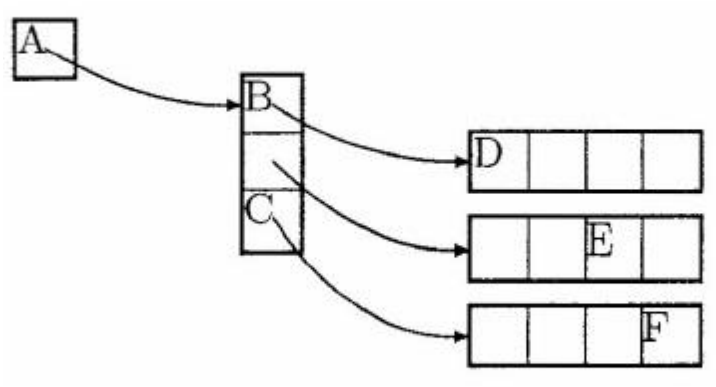
ICS4UO - Exercise – Multi-Dimensional Array

1. **The diagram shows an array declared by the statement.**

```
int[][] a = new int[3][4];
```

State the identifier of each cell marked by a letter.



```
A - a[0][0];
B - a[0][0];
C - a[3][0];
D – a[0][0];
E – a[1][2];
F – a[2][3];
```

2. **How many elements would there be in each of the arrays created by the following declarations?**

(a) `double[][] first = new double[25][40];`

```
There would be 1000 elements.
```

(b) `boolean[][][] second = new boolean[3][6][50];`

```
There would be 900 elements.
```

(c) `char[][] third = new char[60][40];`

```
There would be 2400 elements.
```

(d) `long[][][] fourth = new long[5][10][20];`

3. Suppose that the following declarations have been made:

```
int a[][] = {{4,2,7},
             {3,9,1}};
int i,j;
```

Determine what would be printed by each fragment.

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

```
OUTPUT:

427

391
```

b.
```
for(i = 0; i < a[0].length; i++)
{
  for(j = 0; j < a.length; j++)
    System.out.print(a[j][i]);
    System.out.println();
}
```

```
OUTPUT:

43

29

71
```

c.
```
for(i = a.length - 1; i >= 0; i--)
{
  for(j = 0; j < a[0].length; j++)
    System.out.print(a[i][j]);
  System.out.println();
}
```

```
OUTPUT:

391
```

```
427
```

4. For the array given in the previous question, write a fragment that would print the elements of the array in the form.

```
17
92
34
```

```
            int a[][] = {{4,2,7},
                         {3,9,1}};
            int i,j;

            for (j = a[0].length - 1; j >= 0; j--) {
                  for (i = a.length - 1; i >=0; i--) {
                        System.out.print(a[i][j]);
                  }
                  System.out.println();
               }
```

5. Write a method sumhaving one `double[][]` parameter. The method should return the sum of the elements of the array passed to it. You may assume that the array is rectangular.

```
      public static void main(String[] args) {
            // TODO Auto-generated method stub
            double [] [] parameter = {{1,1,1,1},{1,1,1,1}};
            System.out.println(sumhaving(parameter));

         }

         public static double sumhaving (double[][] parameter) {
            double sum = 0;
            for (int x = 0; x < parameter.length; x++) {
                  for (int y = 0; y < parameter[x].length; y++) {
                        sum = sum + parameter[x][y];
                  }
            }
            return sum;

         }
```

6. Write a method `max` that will return the maximum value of the elements in a two-dimensional array of int values. Do *not* assume that the array is rectangular.

```
                int [][] parameter = {{1,2,3,4}, {5,6,7,8}};
                System.out.println(max(parameter));

        }

        public static int max (int [][] parameter) {
                int max = 0;
                for (int x = 0; x < parameter.length; x++) {
                        for (int y = 0; y < parameter[x].length; y++) {
                                if (max < parameter[x][y])
                                        max = parameter[x][y];
                        }
                }
                return max;

        }
```

7. Write a method `print` that could be used to print a two-dimensional ragged array of `int` values. Each row of elements should be printed on its own line with one blank between each element.

```
                int [][] parameter = {{1,2,3}, {4,5,6}};
                print(parameter);
        }

        public static void print (int [][] parameter) {
                for (int x = 0; x < parameter.length; x++) {
                        for (int y = 0; y < parameter[x].length; y++) {
                                System.out.print(parameter[x][y] + "\t");
                        }
                        System.out.println();
                }

        }
```

8. Write a method `size` that has one `int[][][]` parameter. The method should return the number of elements in the array. Do not make any assumptions about regularity of the array.

```
        public static void main(String[] args) {
                // TODO Auto-generated method stub
                int [][][] parameter = new int [4][4][4];
                //Sets all the elements of the array to 1
                for (int x = 0; x < parameter.length; x++) {
                        for (int y = 0; y < parameter[x].length; y++) {
                                for (int z = 0; z < parameter[x][y].length;
    z++) {
```

```
                    parameter[x][y][z] = 1;
                }
            }
        }
        System.out.println(size(parameter));

    }

    public static int size (int [][][] parameter) {
        int size = 0;
        for (int x = 0; x < parameter.length; x++) {
            for (int y = 0; y < parameter[x].length; y++) {
                for (int z = 0; z < parameter[x][y].length;
 z++) {
                    size++;
                }
            }
        }
        return size;

    }
```