COMMONS OF ARRAY IN JAVA

- ***Tính tổng giá trị các phần tử 2 chiều trong mảng theo ForEach***:

```
public int sumOfIntegerUseForEach(int[][] array) {
    //= new int[hàng][cột];
        int sum = 0;
    for (int[] row : array) {
      for (int column : row) {
         sum += column;
      }
    }
    return sum;
  }
```

- ***Tính tổng giá trị các phần tử 2 chiều trong mảng theo For:***

```
public int sumOfIntegerUseFor(int[][] array) {
    int sum = 0;
    for (int row = 0; row < array.length; row++) {
      for (int column = 0; column < array[row].length; column++) {
         sum += array[row][column];
      }
    }
    return sum;
  }
```

- ***Tính tổng giá trị các phần tử trong mảng theo đường chéo***:

```
public int sumOfMajorDiagonal(int[][] array) {
    int sum = 0;
    for (int row = 0; row < array.length; row++) {
      for (int column = 0; column < array[row].length; column++) {
        if (row == column) {
          sum += array[row][column];
        }
      }
    }
    return sum;
  }
```

- ***Sắp xếp theo kiểu Selection Sort:***

```
public int[] SelectionSort(int[] num) {
    int i, j, first, temp;
    for (i = num.length - 1; i > 0; i--) {
      first = 0;   //initialize to subscript of first element
      for (j = 1; j <= i; j++) //locate smallest element between positions 1 and i.
      {
        if (num[ j] < num[ first]) {
          first = j;
        }
      }
      temp = num[ first];   //swap smallest found with element in position i.
      num[ first] = num[ i];
      num[ i] = temp;
```

```
    }
    return num;
  }
```

- ***Sắp xếp theo kiểu Insertion Sort:***

```
public int[] InsertionSort(int[] num) {
    int j;    // the number of items sorted so far
    int key;  // the item to be inserted
    int i;
    for (j = 1; j < num.length; j++) // Start with 1 (not 0)
    {
      key = num[j];
      for (i = j - 1; (i >= 0) && (num[i] < key); i--) // Smaller values are moving
      {
        num[i + 1] = num[i];
      }
      num[i + 1] = key;   // Put the key in its proper location
    }
    return num;
  }
```

- ***Duyệt và in giá trị trong mảng 1 chiều:***

```
public void printArray(int[] array) {
    for (int i : array) {
      System.out.print(i + " ");
    }
```

```
}
```

- ***Duyệt và in giá trị trong mảng 2 chiều:***

```
public void printArray(String[][] array) {
    for (int row = 0; row < array.length; row++) {
        for (int column = 0; column < array[row].length; column++) {
            System.out.print("\t" + array[row][column] + " ");
        }
        System.out.println("\n");
    }
}
```

- ***Tính tổng các giá trị phần tử trong mảng 2 chiều:***

```
public double SumElements(double[][] arr){
    double total=0;
    for(int row = 0;row<arr.length;row++){
        for(int column = 0;column<arr[row].length;column++){
            total+=arr[row][column];
        }
    }
    return total;
}
```

- ***Tính tổng các giá trị phần tử theo cột:***

```
public void SumElementsByColumn(double[][] arr){
    for(int column = 0;column<arr[0].length;column++){
        double total=0;
```

```
        for(int row = 0;row<arr.length;row++){
            total += arr[row][column];
            System.out.println("arr["+row+"]["+column+"] = "+ total);
        }
    }
}
```

- ***Tính tổng các giá trị phần tử theo hàng:***

```
public void SumElementsByRow(double[][] arr){
    for(int row = 0;row<arr[0].length;row++){
        double total=0;
        for(int column = 0;column<arr.length;column++){
            total += arr[row][column];
            System.out.println("arr["+row+"]["+column+"] = "+ total);
        }
    }
}
```

- ***Tìm row chứa giá trị tổng số lớn nhất:***

```
public int SumRowLargest(double[][] arr){
    int maxRow = 0;
    int indexOfMaxRow = 0;
    // Get sum of the first row in maxRow
    for (int column = 0; column < arr[0].length; column++) {
```

```
    maxRow += arr[0][column];
  }
  for (int row = 1; row < arr.length; row++) {
    int totalOfThisRow = 0;
    for (int column = 0; column < arr[row].length; column++) {
      totalOfThisRow += arr[row][column];
    }
      if (totalOfThisRow > maxRow) {
        maxRow = totalOfThisRow;
        indexOfMaxRow = row;
      }
  }
 Return  maxRow;
}
```

--- Hết ---