```
Sorting Array
                                                                  نطبع <del>ق</del>ِل الترتيب 🛴
                                                                                                                                -> main method
    stem.out.println();
                                                                                                                                                bubble sort
  ic static void selectionSort(int[] a) {
int length = a.length;
for (int i = 0; i < length - 1; i++) {
   int min = i;
   for (int j = i + 1; j < length; j++) {
        if (a[j] < a[min]) {
            min = j;
        }
}</pre>
                                                                                                                          > Selection sort + swap method
      swap(a, min, i);
lic static void insertionSort(int[] ;
int length = a.length;
for (int i = 1; i < length; i++) {
   int temp = a[i];
   int | = i;</pre>
                                                                                                                          insertion Sort
      // shift and decrement j
while (j > 0 && a[j - 1] > temp) {
    a[j] = a[j - 1];
}
     )
a[j] = temp;
 ic static void quickSort(int[] a) {
quickSort(a, 0, a.length - 1);
 iic static void quickSort(int[] a, int min, int max) {
   if (min < max) {
      int index = partition(a, min, max); // get pivot a
      quickSort(a, min, index - 1); // sort left
      quickSort(a, min, index + 1, max); // sort right
   }</pre>
lic static int partition(int[] a, int min, int max) {
  int middle = (min + max) / 2;
  int pivot = a[middle];
                                                                                                                                                                                                 quick sort + swap method
int left = min;
int right = max;
      // 1. move left
while (left < right && (a[left] <= pivot)) {
    left++;</pre>
      // 2. move right
while (a[right] > pivot) (
    right--;
      if (left < right) (
    swap (a, left, right);</pre>
                                                                                                                                                                                           merge sort
   compare between elements in the two subbrrays are
hile (first 10 alst 16 first2 0 alst2) {
  if (affirst1) < affirst2)) {
    temp(k) = affirst2);
    first1++;
  } else {
    temp(k) = affirst2);
    first2++;
    f insert remaining eleme
sile (first2 <= last2) {
  temp[k] = a[first2];
  first2++;
  k++;</pre>
 // copy back to old array

for (k = first; k <= last; k++) {
    a[k] = temp[k];
```