***AP Java Sorting worksheet 2 Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Questions 1 - 3 refer to the following program which correctly sorts the elements of **nums** into ascending order :

**public void setup() {**

**int [] nums = {3,-1, 2, 5,-3};**

**mysterySort(nums);**

**for (int i : nums)**

**System.out.print(i+", ");**

**}**

**public static void mysterySort(int[] items) {**

**for (int outer = 1; outer < *items.length*; outer++)**

**{**

**int position = outer;**

**int k = items[position];**

**// Shift larger values to the right**

**while (position > 0 && items[position - 1] > k)**

**{**

**items[position] = items[position - 1];**

**position--;**

**}**

**items[position] = k;**

***/\* end of* for *loop \*/***

**}**

**}**

1. The sorting algorithm implemented in the sort method can be best described as

A. Selection sort B. Insertion sort C. Bubble sort

2. What would be the order after 3 passes of the for loop (i.e. when **outer**=3 at the point indicated by ***/\* end of* for *loop \*/*** )?

3. Change one line of code of the sort method so the program correctly sorts the integers in **nums** into ***descending*** order.

Questions 4 - 6 refer to the following program which correctly sorts the elements of **nums** into ascending order :

**public void setup() {**

**int [] nums = {3,-1, 2, 5,-3};**

**mysterySort(nums);**

**for (int i : nums)**

**System.out.print(i+", ");**

**}**

**public static void mysterySort(int[] items) {**

**for (int outer = 0; *outer < items.length - 1*; outer++)**

**{**

**for (int inner = 0; inner < items.length-outer-1;**

**inner++)**

**{**

**if (items[inner] > items[inner + 1])**

**{**

**//swap list[inner] & list[inner+1]**

**int temp = items[inner];**

**items[inner] = items[inner + 1];**

**items[inner + 1] = temp;**

**}**

**}**

***/\* end of outer* for *loop \*/***

**}**

**}**

4. The sorting algorithm implemented in the sort method can be best described as

A. Selection sort B. Insertion sort C. Bubble sort

5.What would be the order after 2 passes of the for loop (i.e. when **outer**=1 at the point indicated by ***/\* end of outer* for *loop \*/*** )?

6. Change one line of code of the sort method so the program correctly sorts the integers in **nums** into ***descending*** order.

Questions 7-9 refer to the following program which correctly sorts the elements of **nums** into ascending order :

**public void setup() {**

**ArrayList <Integer> nums = new ArrayList <Integer>();**

**nums.add(3);**

**nums.add(-1);**

**nums.add(2);**

**nums.add(5);**

**nums.add(-3);**

**mysterySort(nums);**

**System.out.println(nums);**

**}**

**public static void mysterySort(ArrayList <Integer> items) {**

**int f, temp;**

**for (int outer = 0; outer < items.size() - 1; outer++)**

**{**

**f = outer;**

**for (int inner = outer + 1; inner < items.size(); inner++)**

**{**

**if (items.get(inner) < items.get(f))**

**{**

**f = inner;**

**}**

**}**

**//swap list.get(outer) & list.get(f)**

**temp = items.get(outer);**

**items.set(outer,items.get(f));**

**items.set(f,temp);**

***/\* end of outer* for *loop \*/***

**}**

**}**

7. The sorting algorithm implemented in the sort method can be best described as

A. Selection sort B. Insertion sort C. Bubble sort

8. What would be the order after 2 passes of the for loop (i.e. when **outer**=1 at the point indicated by ***/\* end of outer* for *loop \*/*** )?

9. Change one line of code of the sort method so the program correctly sorts the integers in **nums** into ***descending*** order.