26.5 Problem Set Worksheet.

Sort the following elements using Selection Sort. Show each pass.

 

1st pass : 6-2-A-3-4

2nd pass : A-6-2-3-4

3rd pass: A-2-6-3-4

4th pass: A-2-3-6-4

5th pass: A-2-3-4-6

Write the Code for each method using the previous exercise.

|  |
| --- |
| class Program {  //INSTANCE VARIABLES NOT SHOWN   public static void main(String[] args)  {  selectionSort(...);   }    public static void selectionSort(int[] array){   for (int i = 0; i < array.length; i++) {  int minValue = findMinimum(array, i)  swap(array, i, minValue)  }   return(array);   }  public static int findMinimum(int[] array, int first){    int min;    for (int x = first; x < array.length; x++) {  if(array[x] < min)  min = array[x];  }  return min;  }        public static void swap(int[] array, int x, int y){    int temp = array[x];  array[x] = array[y]  array[y] = temp;    }     } |

Sort the following elements using Bubble Sort. Show each pass.



1st pass : 5-7-2-1

2nd pass :5-7-2-1

3rd pass: 5-2-7-1

4th pass: 5-2-1-7

5th pass: 2-5-1-7

6th pass: 2-1-5-7

7th pass: 1-2-5-7

Write the code for the previous exercise.

|  |
| --- |
| class Program { // INSTANCE VARIBLES NOT SHOWN    public static void main(String[] args)  {   BubbleSort(...);    }    public static void BubbleSort(int[] a){    int k = 0;  boolean swapOccured = true;   while ((k < array.length() - 1) && swapOccured){  swapOccured = false;  k++;    for (int j = 0; j < array.length() - k; j++)   if (array[j].compareTo(array[j + 1]) > 0) {  swap(array, j, j+1)  swapOccured = true;  }  }      }      public static void swap(int[] array, int x, int y){    int temp = array[x];  array[x] = array[y]  array[y] = temp;   }     } |

Sort the following elements using Insertion Sort.



1st pass : 1-4-2-5-3

2nd pass : 1-4-2-5-3

3rd pass: 1-4-2-5-3

4th pass: 1-2-4-5-3

5th pass: 1-2-4-5-3

6th pass: 1-2-4-3-5

7th pass: 1-2-3-4-5

Write the code for the previous exercise.

|  |
| --- |
| class Program {  //INSTANCE VARIABLES NOT SHOWN   public static void main(String[] args)  {   insertionSort(a);    }    public static void insertionSort(int[] array){    for (int i = 1; i < array.length; i++) {  int k = array[i];  ink j = i - 1;    while (j >= 0 && array[j] > key)  {  array[j + 1] = array[j];  }  array[j + 1] = key;  }    }   } |