**Learning by doing-2**

Show, how Insertion sort sorts the array **E A S Y Q U E S T I O N**.

* Answer the above problem in a file name problem-2.txt and add it to the folder ADS-1-practice/m8
* Push the respository to GitHub when your answer is done
* Enter the git commit ID in the blank below

Do not use eval to submit this activity

### Activity6

write the missing statements in the below code:\

**public class Selection {**

**public static void sort(Comparable[] a) {**

**int N = ---------------;    //Blank 1**

**for (int i = 0; i < N; i++) {**

**Int min = ---------------;    //Blank 2**

**for (int j = i+1; j < N; j++)**

**if (less(a[j], a[min]))**

**min = ------------------; //Blank3**

**exch(a, i, ----------------);     //Blank4**

**}**

**}**

**}**

### Activity7

**If a = {3, 6, 4, 2, 1, 0} is the input for below code**

**public class Insertion {**

**public static void sort(Comparable[] a) {**

**int N = a.length;**

**for (int i = 1; i < N; i++) {**

**for (int j = i; j > 0 && less(a[ j ], a[j-1]); j--)**

**exch(a, j, j-1);**

**}**

**}**

**}**

**}**

**Answer the following questions**

**Note: while writing answers use the following notation [1, 2, 3, 4]**

**Print the elements in the array a, after completion of 2nd iteration of loop i**

**Print the elements in the array a, in 3rd iteration of i and after completion of 2nd iteration of loop j**

**Print the elements in the array a, in 4th iteration of i and after completion of 1st iteration of loop j**

**Learning by doing-3**

Which method runs faster for an array with all keys identical, basic selection sort or basic insertion sort?

* Answer the above problem in a file name problem-3.txt and add it to the folder ADS-1-practice/m8
* Push the respository to GitHub when your answer is done
* Enter the git commit ID in the blank below

Do not use eval to submit this activity

**Learning by doing-4**

Which method runs faster for an array in reverse order, basic selection sort or basic insertion sort?

* Answer the above problem in a file name problem-4.txt and add it to the folder ADS-1-practice/m8
* Push the respository to GitHub when your answer is done
* Enter the git commit ID in the blank below

Do not use eval to submit this activity

**Assignment-1**

**Selection Sort:** Solve the given problem using Selection sort. Check your program against the given test cases below as a zip file. Submit your Solution(zip file) when all the test cases are passed.

* Download the starter code; the directory structure is similar to the sample-assignment seen in the previous activity
* Add a subfolder m8 for Module 8
* Unzip the starter code into m8 folder. You should see a folder with the name 8.1 Team Ranking With Insertion
* You should write your solution in the file Solution.java
* There are a few lines of code to handle the input testcases
* After you write the code use eval to check if you got all the testcases right
* submit commit ID in the textbox below.

**Assignment-2**

**Insertion Sort :** Solve the given problem using Insertion sort. Check your program against the given test cases below as a zip file. Submit your Solution(zip file) when all the test cases are passed.

* Download the starter code; the directory structure is similar to the sample-assignment seen in the previous activity
* Add a subfolder m8 for Module 8
* Unzip the starter code into m8 folder. You should see a folder with the name 8.2 Team Ranking With Selection
* You should write your solution in the file Solution.java
* There are a few lines of code to handle the input testcases
* After you write the code use eval to check if you got all the testcases right
* submit commit ID in the textbox below.

**Bonus problem-1**

**Insertion sort with sentinel:** Develop an implementation of insertion sort that eliminates the j > 0 test in the inner loop by first putting the smallest item into position. Check your program against the given test cases below as a zip file. Submit your Solution(zip file) when all the test cases are passed.

* Download the starter code; the directory structure is similar to the sample-assignment seen in the previous activity
* Add a subfolder m8 for Module 8
* Unzip the starter code into m8 folder. You should see a folder with the name 8.2 Insertion sort with sentinel
* You should write your solution in the file Solution.java
* There are a few lines of code to handle the input testcases
* After you write the code use eval to check if you got all the testcases right
* submit commit ID in the textbox below.