Mariya Eggensperger
CST 370, Spring 2017
Dr. Feiling Jia
Design/Analysis
of Algorithms

CST 370
Programming Assignment (Sorting Algorithms)

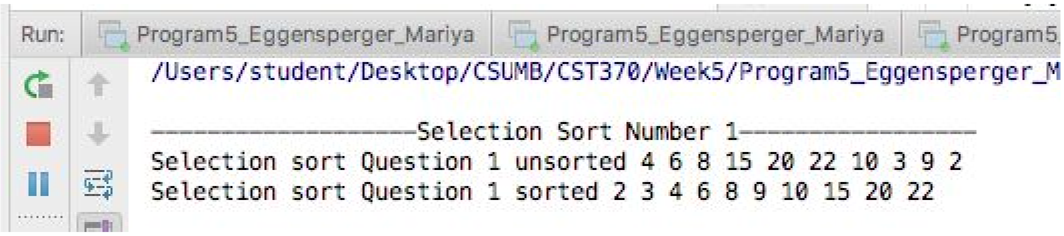
In this assignment you will implement the sorting algorithms covered in class.

1. In the first part of the assignment you will implement the simple selection sort algorithm. You can use an array to store the elements.

As you are aware the selection sort algorithm runs in O(n2) time. Implement your algorithm in a separate function called SelectionSort().

Sample Test Cases
Input: 4, 6, 8, 15, 20, 22, 10, 3, 9, 2
Output: 2, 3, 4, 6, 8, 9, 10, 15, 20, 22

Figure 1 In this image, it is evident that the program is correctly running the SelectionSort() function and sorting the elements in the array in ascending order.



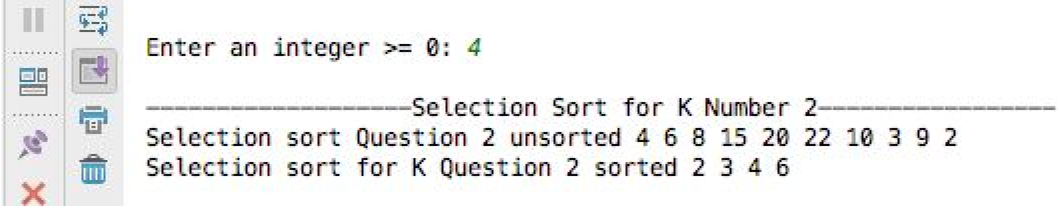
```
/Users/student/Desktop/CSUMB/CST370/Week5/Program5_Eggensperger_M

-------------------Selection Sort Number 1------------------
Selection sort Question 1 unsorted 4 6 8 15 20 22 10 3 9 2
Selection sort Question 1 sorted 2 3 4 6 8 9 10 15 20 22
```

2. In the second part of the assignment you will modify the selection sort algorithm to sort the first k smallest elements of the array (the value of k will be entered by the user). Your algorithm must run in O(nk) time.

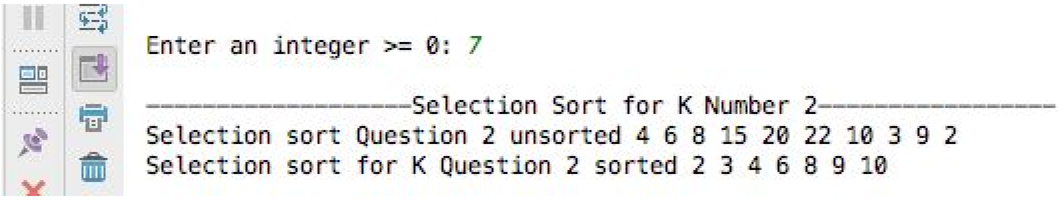
Sample Test Case 1
Input: 4, 6, 8, 15, 20, 22, 10, 3, 9, 2 k=4
Output: 2, 3, 4, 6

Sample Test Case 2
Input: 4, 6, 8, 15, 20, 22, 10, 3, 9, 2 k=6
Output: 2, 3, 4, 6, 8, 9

Figure 3 Given the first k smallest element of the array (k is entered by the user), the program executes the selection sort algorithm in retrospect of the user's k size input, whether it be even or odd.



```
Enter an integer >= 0: 4

-------------------Selection Sort for K Number 2-----------------
Selection sort Question 2 unsorted 4 6 8 15 20 22 10 3 9 2
Selection sort for K Question 2 sorted 2 3 4 6
```

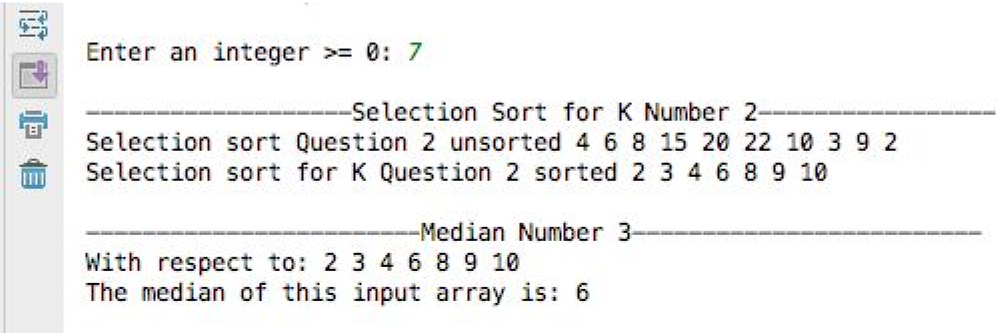


```
Enter an integer >= 0: 7

-------------------Selection Sort for K Number 2-----------------
Selection sort Question 2 unsorted 4 6 8 15 20 22 10 3 9 2
Selection sort for K Question 2 sorted 2 3 4 6 8 9 10
```

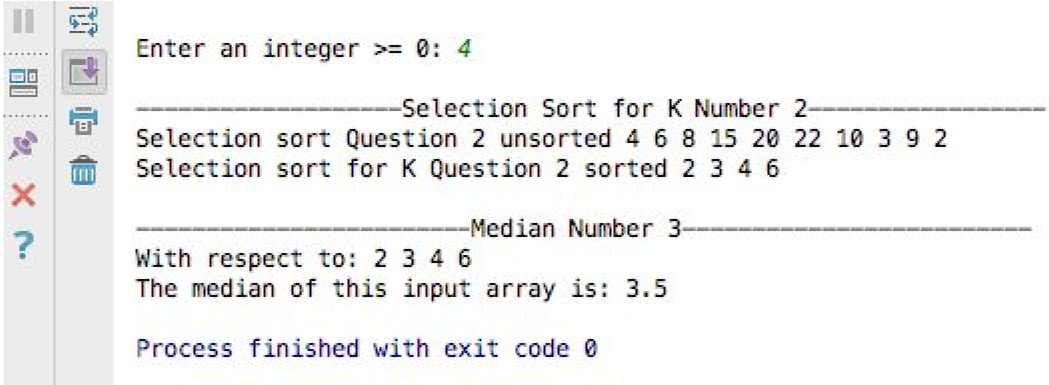
3. Use the implementation in part 2 to determine the median value of the input array. You can assume that the values in the array are distinct. If the array is of even size, then the median is the average of the two middle values is the array.

[Hint: Median is the middle value in the array i.e., it is that value in the array for which exactly half the values are greater than and half the values are less than it]

Figure 3 In respect to k and the implementation in part 2, for Number 3, the function retains the previous selection sort values for k and determines their median. If odd, the function correctly executes down the array until a final element is found as median; if even, the function take the two middle values and divides them by 2.0 for a double return.



```
Enter an integer >= 0: 7

-------------------Selection Sort for K Number 2-----------------
Selection sort Question 2 unsorted 4 6 8 15 20 22 10 3 9 2
Selection sort for K Question 2 sorted 2 3 4 6 8 9 10

------------------------Median Number 3-------------------------
With respect to: 2 3 4 6 8 9 10
The median of this input array is: 6
```



```
Enter an integer >= 0: 4

-------------------Selection Sort for K Number 2-----------------
Selection sort Question 2 unsorted 4 6 8 15 20 22 10 3 9 2
Selection sort for K Question 2 sorted 2 3 4 6

------------------------Median Number 3-------------------------
With respect to: 2 3 4 6
The median of this input array is: 3.5

Process finished with exit code 0
```