Data to test:

We must test three different arrays, while doing testing these three different arrays, we must check if k is in Side 1 (the left), Side 2 (the right), and if k is the pivot (partition) value. The three types of arrays will be tested, is first, an array with integers on both sides of the pivot, secondly, an array with integers all on the right side of the pivot or in terms of this program, Side 1, and lastly, an array with integers all on the left side of the pivot (Side 2).  
  
Array 1:

4 1 3 5 7 9 8

Expected Output:

1, and 3 on S1.  
4 is the pivot.

5, 7, 9, and 8 are on S2.

Array 2:

9 6 4 3 2  
Expected Output:

9 is the pivot.

6, 4, 3, and 1 are on S1.

Array 3:

2 6 8 9 5 7

Expected Output:

2 is the pivot.

6, 8, 9, 5, 7, and 4 are on S2.

User Document

**Introduction:**

This program uses recursion to implement the kSmall algorithm in the textbook Data Abstraction and Problem Solving with C++: Walls and Mirrors, 7e (Ch. 2 Section 4.4). There you will find pseudocode for an almost complete kSmall function. Next, we needed to create a partition function that would partition the user entered array, it will partition the array using the first element as the partition. If the first element in the array is the smallest, all elements will be in S2. If the first element is the largest element in the array, all other elements will be in S1. The overall goal is to implement the recursive solution to the kSmall function.

**File Location:**

The programs name is mainProject3.cpp. It is located in the following directory on CentOS: **/home/STCLOUDSTATE/hp0106rn**

**How To Compile Program:**

Enter: **g++ mainProject3.cpp**

**How To Run Program:**

Enter: **./a.out** to begin the program.

Prompt> “How many integers do you need to enter?”

User Input> User enters the amount of elements (integers) they have.

Prompt> “Please enter …. Integers.”

User Input> User enters the elements of the array separated by a space

Displayed To User> “Here is your partitioned array: “ <PARTITIONED ARRAY>

Prompt> “Enter the index (k) you would like to test: “

User Input> User enters what element they would like to test.

Displayed To User> “The <kth> smallest element is <p>

Displayed To User> “The kth element <p> is in (S1, S2)/is the pivot.

Program then terminates.