**Problem Specification:**

The program needs to have 3 procedures:

1. search: that takes two parameters, a list of numbers and the number to find in that list
2. binary\_search: that takes two parameters, a sorted list of numbers and the number to find
3. insertion\_sort: takes a list of numbers and sort it

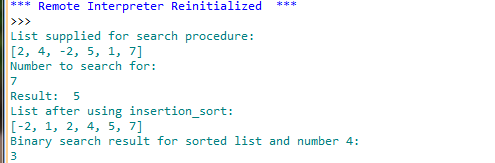
**Testing Plan:**

For “search”, a list of numbers and a number to search for will be supplied and output will be observed. For “binary\_search”, a sorted list and number to search will be supplied. To test “insertion sort”, a list of unsorted numbers will be provided.

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Procedure | Input | Expected Output |
| 1 | 1 | [1, 2, 4, -2, 6], 3 | -1 |
| 2 | 1 | [9, 2, 5, 6, 100], 6 | 3 |
| 3 | 2 | [2, 3, 6, 8, 299], 100 | -1 |
| 4 | 2 | [3, 5, 7, 90], 3 | 0 |
| 5 | 3 | [5 ,3, -7, 5, 1] | [-7, 1, 3, 5, 5] |
| 6 | 3 | [3, 5, 1, -4, 3, 8] | [-4, 1, 3, 3, 5, 8] |

**Sample Execution:**



**Analysis & Conclusion:**

All three procedures of the assignment is written and tested with test-cases. Outputs generated by these procedures are matched with expected ones and provides positive matches.

The code is submitted and report is written based on submitted code. The report format supplied as a reference has been followed.

**References:**

Insertion sort,

<http://en.wikipedia.org/wiki/Insertion_sort>.

**Feedback:**

The assignment was not very difficult to complete. It covered basic materials of programming using python, writing procedures and analyzing complexity of algorithms.

**Test log:**

|  |  |  |
| --- | --- | --- |
| No. | Output | Result Status |
| 1 | -1 | Pass |
| 2 | 3 | Pass |
| 3 | -1 | Pass |
| 4 | 0 | Pass |
| 5 | [-7, 1, 3, 5, 5] | Pass |
| 6 | [-4, 1, 3, 3, 5, 8] | Pass |