**Test Case Template**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Name:** | | **Astronomical Processing** | | | | | | | | | |
| **Test Scenario No:** | | 1 | | **Test Scenario Name:** | | | Final Test | | | | |
| **Description:** | | Test final Application for all possible scenarios | | | | | | | | | |
| **Developer:** | | C Blunt | | | **Tester:** | *Bruce Fisher P197681* | | | **Date:** | 19/09/2021 | |
| **Test Case No** | **Test Case Name** | | **Test steps** | | | | **Test Data** | **Expected result** | | | **Pass / Fail** |
| 1 | Auto Fill Data | | 1. Click on AutoFill button. | | | | Random generated by AutoFill Method | Random Data fills array with random data in the range 10 to 99.  Reports to user with message that the operation completed. | | | Pass |
| 2 | Sort Data in List. | | 1. Click on Sort button. | | | | Data already in the list | Sorts data in ascending order and displays sorted data back in list.  Reports to user with message that operation completed. | | | Pass |
| 3 | Clicking on data from list display the value in the Text Box. | | 1. Click on value from the list. | | | | Selected from the list of Data | Displays the selected value from the list in the Text Box. | | | Pass |
| 4 | Add Data to Full List. | | 1. List already Full from Auto Fill. 2. Type value into Text Box. 3. Click on Add button. | | | | 10 | Displays error message stating the list is already full and clears the value displayed in the Text Box. | | | Fail |
| 5 | Add Data to Full List – bug fixed see version report. | | 1. List already Full from Auto Fill. 2. Type value into Text Box. 3. Click on Add button. | | | | 10 | Displays error message stating the list is already full and clears the value displayed in the Text Box. | | | Pass |
| 6 | Delete Data from List to allow value to be Added. | | 1. Click on Value from List to Delete. 2. Click on Delete button. 3. User prompt to confirm delete click on NO. | | | | Selected from the list of Data. | Displays selection in the Text Box and prompts user to confirm delete yes/no.  Once no clicked does not delete the value and displays message to confirm the process has completed and clears the Text Box. | | | Fail |
| 7 | Delete Data from List to allow value to be Added – bug fixed see version report.  Further testing for user mistake clicking on Delete again before selection from list does not delete previous selection. | | 1. Click on Value from List to Delete. 2. Click on Delete button. 3. User prompt to confirm delete click on NO. | | | | Selected from the list of Data. | Displays selection in the Text Box and prompts user to confirm delete yes/no.  Once no clicked does not delete the value and displays message to confirm the process has completed and clears the Text Box.  Does not allow the user to again press delete with no new selection from list. | | | Fail |
| 8 | Delete Data from List to allow value to be Added – bug fixed see version report. | | 1. Click on Value from List to Delete. 2. Click on Delete button. 3. User prompt to confirm delete click on NO. | | | | Selected from the list of Data | Displays selection in the Text Box and prompts user to confirm delete yes/no.  Once NO clicked does not delete the value and displays message to confirm the process has completed and clears the Text Box.  Does not allow the user to again press delete with no new selection from list. | | | Pass |
| 9 | Delete Data from List to allow value to be Added. | | 1. Click on Value from List to Delete. 2. Click on Delete button. 3. User prompt to confirm delete click on YES. | | | | Selected from the list of Data | Displays selection in the Text Box and prompts user to confirm delete yes/no.  Once YES clicked deletes the value and displays message to confirm the process has completed and clears the Text Box.  Does not allow the user to again press delete with no new selection from list. | | | Pass |
| 10 | Add Data to List. | | 1. Type value into Text Box. 2. Click on Add button. | | | | 10 | Adds the value to the list and redisplays the amendment for list. Does not allow user to click Add button again. | | | Pass |
| 11 | Add Data to List no input in Text Box. | | 1. Click on Add button. | | | | nothing | Displays error message prompting user to enter value in the Text Box. | | | Pass |
| 12 | Add invalid Data to list. | | 1. Type invalid value into Text Box. 2. Click on Add button. | | | | A1 | Displays error message prompting user to enter valid integer value in the Text Box. Clears out invalid entry from Text Box. | | | Pass |
| 13 | Add invalid Data to list. | | 1. Try to type in data larger than 2 digits into Text Box. | | | | 100 | Will not allow further input. | | | Pass |
| 14 | Edit Data from List. | | 1. Select value from list and edit in Text Box. 2. Click on Edit button. | | | | Selected from the list of Data.  Changed to 10. | Will display selected value in Text Box and allow to be edited and upon clicking Edit button changes value in the displayed list. Notifies user the change has been processed. | | | Pass |
| 15 | Edit Data from List but try to edit with invalid entry and longer then 2 digits. | | 1. Select value from list and edit in Text Box. 2. Click on Edit button. | | | | Selected from the list of Data.  Changed to A10. | Will display error message prompting user to enter a valid integer and does not allow 3 digits to be entered. Clears invalid entry from Text Box. | | | Pass |
| 16 | Edit Data from List but after last process just click on Edit button again to check does not contain previous entry and user has to make selection from list for value to edit. | | 1. Click on Edit button again. | | | | nothing | Will display error message prompting user to select value from list first. | | | Pass |
| 17 | Binary search before sorting the data after changes have been made to list as per above. | | 1. Click on Search button. | | | | nothing | Will display error message prompting user to Sort the list first. | | | Pass |
| 18 | Binary search after sorting the data with nothing in the Text Box to search. | | 1. Click on Sort button. 2. Click on Search button. | | | | nothing | Will display error message prompting user to enter value in the Text Box to search. | | | Pass |
| 19 | Binary search after sorting with invalid value in Text Box to search for. | | 1. Click on Sort button. 2. Enter INVALID value into Text Box. 3. Click on Search button | | | | 1A | Will display error message prompting user to enter a valid integer. Clears out the invalid data from the Text Box. | | | Pass |
| 20 | Binary search for data from value in the Text Box that exists in the current list. | | 1. Enter value in the Text Box. 2. Click on the Search button. | | | | 85  Data in List  10, 10, 10, 10, 10, 14, 25, 30, 33, 34, 36, 42, 43, 44, 46, 46, 57, 57, 61, 65, 67, 85, 86, 95 | Displays message that the value 85 was found at index 22 in the list and clears out the selection. | | | Pass |
| 21 | Binary search for data again straight after above and just click button again. | | 1. Click on the Search button. | | | | nothing | Displays message prompting user to enter value in the Text Box. | | | Pass |
| 22 | Binary search for data from value in the Text Box that does NOT exist in the current list. | | 1. Enter value in the Text Box. 2. Click on the Search button. | | | | 11  Data in List  10, 10, 10, 10, 10, 14, 25, 30, 33, 34, 36, 42, 43, 44, 46, 46, 57, 57, 61, 65, 67, 85, 86, 95 | Displays message that the value 11 was NOT found in the list and clears out the selection. | | | Pass |
| 23 | Binary search for data from value in Text Box that exist in the list that has DUPLICATES. | | 1. Enter value that is duplicated in the list. 2. Click the Search button. | | | | 10  Data in List  10, 10, 10, 10, 10, 14, 25, 30, 33, 34, 36, 42, 43, 44, 46, 46, 57, 57, 61, 65, 67, 85, 86, 95 | Ideally finds the first value in the list from Binary search. | | | Fail |
|  |  | |  | | | |  | Finds 3rd value duplicate in the list. Failed as a Binary Search cannot find first value in a duplicate list depends on surrounding values what index it finds first. | | |  |
| 23 | NEW Binary search that’s handles SINGLE & DUPLICATE entrees in a list of data. Test for duplicates. | | 1. Enter value that is duplicated in the list. 2. Click the Search button. | | | | 10  Data in List  10, 10, 10, 10, 10, 14, 25, 30, 33, 34, 36, 42, 44, 44, 44, 46, 57, 57, 61, 65, 67, 95, 95, 95 | Finds 5 and lists indexes found from first 1 to last 5 and places list on first found. And outputs user message Found 10 Multiples at index 1 to 5. | | | Pass |
| 24 | NEW Binary search that’s handles SINGLE & DUPLICATE entrees in a list of data. Test for single. | | 1. 1 Enter value that is NOT duplicated in the list. 2. Click the Search button. | | | | 36  Data in List  10, 10, 10, 10, 10, 14, 25, 30, 33, 34, 36, 42, 44, 44, 44, 46, 57, 57,  61, 65, 67, 95, 95, 95 | Finds 1 entry in list and places list on that entry and outputs user message Found 36 at index 11. | | | Pass |
| 25 | User input to Text Box within range 10 to 99 only | | 1. Enter value into Text Box for each button below clicked. 2. Click Add, Edit, Delete, and Search for each entry above. | | | | 0  1  01  9  09  10 | Display message for user to enter value within the range 10 to 99.  Last test Data should be valid entry. | | | Fail  Last entry  Pass |
| 26 | User input to Text Box within range 10 to 99 only. Added – bug fixed see version report. | | 1. Enter value into Text Box for each button below clicked. 2. Click Add, Edit, Delete, and Search for each entry above. | | | | 0  1  01  9  09  10 | Display message for user to enter value within the range 10 to 99.  Last test Data should be valid entry. | | | Pass |

Researched Binary Search and confirms that it does not deal with duplicates well see reference below…

<https://en.wikipedia.org/wiki/Binary_search_algorithm>

“The procedure may return any index whose element is equal to the target value, even if there are duplicate elements in the array. For example, if the array to be searched was [1,2,3,4,4,5,6,7]} and the target was 4, then it would be correct for the algorithm to either return the 4th (index 3) or 5th (index 4) element. The regular procedure would return the 4th element (index 3) in this case. It does not always return the first duplicate (consider [1,2,4,4,4,5,6,7]} which still returns the 4th element). However, it is sometimes necessary to find the leftmost element or the rightmost element for a target value that is duplicated in the array. In the above example, the 4th element is the leftmost element of the value 4, while the 5th element is the rightmost element of the value 4. The alternative procedure above will always return the index of the rightmost element if such an element exists.”

*The new Binary Search Algorithm now deals with single and multiple entries.*