#	Pri.	Story	Example	Test
| 1 | | As a user, I want to be able to print out a 1 dimensional array of integers so I can see the results of my array algorithms. | Enter a "p and the calculator prints "{1, 2, 3}". The array is defined in the main program as "float test_array[3] = {1, 2, 3}". | ```
arrays> ./arrays
enter 'x' to exit
Enter command> p
(1.000000, 2.000000, 3.000000)
Enter command>
``` |
| 2 | | As a user, I want to enter a 1-dimensional array of integers so I can sort on them. | If you enter an 'r' command you are prompted for an integer that represents the length of the array. You are then prompted for that number of floating point numbers. The program then prints out the sequence entered. | ```
Enter command> p
(1.000000, 2.000000, 3.000000)
Enter command> r
Enter array length> 3
value 0> > 4
value 1> > 5
value 2> > 6
Enter command> p
(4.000000, 5.000000, 6.000000)
``` |
| 3 | | As a user, I want to be able to read and write a 1 dimensional array in a file so I can save and retrieve the array. | If you enter an 's' command it will save the contents of the current array in a file called "test"; if you enter an 'l' command it will read the contents of the file called "test" into the array. | ```
Enter command> p
(1.000000, 2.000000, 3.000000)
Enter command> r
Enter array length> 3
value 0> > 4
value 1> > 5
value 2> > 6
Enter command> p
(4.000000, 5.000000, 6.000000)
Enter command> s
Enter command> r
Enter array length> 3
value 0> > 1
value 1> > 2
value 2> > 3
Enter command> p
(1.000000, 2.000000, 3.000000)
Enter command> l
Enter command> p
(4.000000, 5.000000, 6.000000)
``` |
| 4 | | As a programmer, I want to add an item to a 1-d array to learn how to do it. | When you enter an 'i' command, you will be prompted for a value to enter and a location at which to put the value. After entering these two numbers, the array will be modified to contain the new value at the specified location. | ```
Enter command> p
(1.000000, 2.000000, 3.000000)
Enter command> i
Enter value to insert> 5
Enter position for value> 1
Enter command> p
(1.000000, 5.000000, 2.000000, 3.000000)
``` |
| 5 | | As a programmer, I want to delete an item from a 1-d array so I can learn how to do it. | When you enter a 'd' command, you will be prompted for a location from which to delete the value. The array will then be modified so that the value at the location specified is removed. | ```
Enter command> p
(1.000000, 5.000000, 2.000000, 3.000000)
Enter command> d
Enter position to delete> 1
Enter command> p
(1.000000, 2.000000, 3.000000)
``` |

6		As a user, I want to print out a 2 dimensional array, so I can see the results of array multiplication.	If you enter 'P', the program will print out a two-dimensional test array stored in the program.	Enter command> P (1.000000, 2.000000, 3.000000) (4.000000, 5.000000, 6.000000) (7.000000, 8.000000, 9.000000)
7		As a user, I want to be able to input out a 2 dimensional array of integers so I can input arrays to multiply.	If you enter an 'R', you are prompted for two integers representing the number of columns and number of rows in the array. You are then prompted for the values for each of the arrays.	Enter command> P (1.000000, 2.000000, 3.000000) (4.000000, 5.000000, 6.000000) (7.000000, 8.000000, 9.000000) Enter command> R Enter 2D array number of columns> 3 Enter 2D array number of rows> 3 value 0> > 9 value 1> > 8 value 2> > 7 value 0> > 6 value 1> > 5 value 2> > 4 value 0> > 3 value 1> > 2 value 2> > 1 Enter command> P (9.000000, 8.000000, 7.000000) (6.000000, 5.000000, 4.000000) (3.000000, 2.000000, 1.000000)
8		As a user, I want to be able to multiply matrices, to show I can manage two dimensional arrays.	If you enter a '*', you are prompted to enter two two-dimensional arrays. The program then prints out the result of matrix multiplication of the arrays.	Enter command> * Enter number of columns for matrix A> 3 Enter number of rows for matrix A> 2 Enter number of columns for matrix B> 2 value 0> > 1 value 1> > 2 value 2> > 3 value 0> > 4 value 1> > 5 value 2> > 6 value 0> > 7 value 1> > 8 value 0> > 9 value 1> > 10 value 0> > 11 value 1> > 12 (58.000000, 64.000000) (139.000000, 154.000000)
9		As a user, I want to be able to do a bubble sort so I can compare the speed of various types of sort.	If you enter an 'S' command you are prompted to enter the type of sort. When you enter 'b' the array entered earlier is then sorted using a bubble sort.	Enter command> p (3.000000, 1.000000, 0.000000, 4.000000, 2.000000) Enter command> S Enter sub-command (b, s, i, m, or q)> b Enter command> p (0.000000, 1.000000, 2.000000, 3.000000, 4.000000) Enter command>

10		As a user, I want to be able to do a selection sort so I can compare the speed of various types of sort.	If you enter an 'S' command you are prompted to enter the type of sort. When you enter 's' the array entered earlier is then sorted using a selection sort.	`Enter command> p`
				`(3.000000, 1.000000, 0.000000, 4.000000, 2.000000)`
				`Enter command> S`
				`Enter sub-command (b, s, i, m, or q)> s`
				`Enter command> p`
				`(0.000000, 1.000000, 2.000000, 3.000000, 4.000000)`

11		As a user, I want to be able to do a insertion sort so I can compare the speed of various types of sort.	If you enter an 'S' command you are prompted to enter the type of sort. When you enter 'i' the array entered earlier is then sorted using a insertion sort.	`Enter command> p`
				`(3.000000, 1.000000, 0.000000, 4.000000, 2.000000)`
				`Enter command> S`
				`Enter sub-command (b, s, i, m, or q)> i`
				`Enter command> p`
				`(0.000000, 1.000000, 2.000000, 3.000000, 4.000000)`
				`Enter command>`

12		As a user, I want to be able to do a linear search so I can compare the speed of linear and binary searching.	If you enter an 'F' command you are prompted to enter the type of search. When you enter 'l' the array entered earlier is then searched using a linear search.	`Enter command> p`
				`(0.000000, 1.000000, 2.000000, 3.000000, 4.000000)`
				`Enter command> F`
				`Search for> 3`
				`Enter sub-command (l or b)> l`
				`Found 3.000000, at position 3`

13		As a user, I want to be able to do a binary search so I can compare the speed of linear and binary searching.	If you enter an 'F' command you are prompted to enter the type of search. When you enter 'b' the array entered earlier is then searched using a binary search.	`Enter command> p`
				`(0.000000, 1.000000, 2.000000, 3.000000, 4.000000)`
				`Enter command> F`
				`Search for> 3`
				`Enter sub-command (l or b)> b`
				`Found 3.000000, at position 3`

14		As a user, I want to be able to do a merge sort so I can compare the speed of various types of sort.	If you enter an 'S' command you are prompted to enter the type of sort. When you enter 'm' the array entered earlier is then sorted using a merge sort.	`Enter command> p`
				`(3.000000, 1.000000, 0.000000, 4.000000, 2.000000)`
				`Enter command> S`
				`Enter sub-command (b, s, i, m, or q)> m`
				`Enter command> p`
				`(0.000000, 1.000000, 2.000000, 3.000000, 4.000000)`

15		As a user, I want to be able to do a quick sort so I can compare the speed of various types of sort.	If you enter an 'S' command you are prompted to enter the type of sort. When you enter 'q' the array entered earlier is then sorted using a quick sort.	`Enter command> p`
				`(3.000000, 1.000000, 0.000000, 4.000000, 2.000000)`
				`Enter command> S`
				`Enter sub-command (b, s, i, m, or q)> b`
				`Enter command> p`
				`(0.000000, 1.000000, 2.000000, 3.000000, 4.000000)`