Pseudocode

display()

print formatted header with column labels for Number, Name, and Party

for every position in myPresidents array

print attributes of each object at current position

// end for

// end displayCountries()

bubbleSortOnNumber()

declare integer *out*

declare integer *in*

for (*out* = number of array elements - 1 to *out* greater than 1, decrement *out*)

for (*in* = 0 to *in* less than *out*, increment *in*)

if (*number* of *in* position in myPresidents array is greater than *number* of *in*+1 position in myPresidents array)

swap (*in*, *in*+1) values

// end if

// end for

//end for

// end bubbleSortOnNumber()

bubbleSortOnName()

declare integer *out*

declare integer *in*

for (*out* = number of array elements - 1 to *out* greater than 1, decrement *out*)

for (*in* = 0 to *in* less than *out*, increment *in*)

if (*name* of *in* position in myPresidents array is greater than *name* of *in*+1 position in myPresidents array)

swap (*in*, *in*+1) values

// end if

// end for

//end for

// end bubbleSortOnName()

swap(int one, int two)

declare Presidents object named *temp* and set to myPresidents[one]

set myPresidents[one] to myPresidents[two]

set myPresidents[two] to *temp*

// end swap()

sequentialSearch(String partySearchKey)

declare integer *count* and set to 0

for(counter i = 0 to i less than length of myPresidents array, increment i)

if (*party* of current position of myPresidents array equals *partySearchKey*)

increment *count*

// end if

// end for

print formatted *partySearchKey*; if count greater than 0, print “Found”, else print “Not Found”; if count greater than 0, print “ occurrences”, else print “”

// end sequentialSearch()

sequentialSearchDisplay()

print formatted title of sequential search

print formatted header with column labels for Search Argument, Result, and Number of Hits

// end sequentialSearchDisplay()

binarySearch(String nameSearchKey)

declare integer *numProbes* and set to 0

declare integer *lowerBound* and set to 0

declare integer *upperBound* and set to number of elements in myPresidents array - 1

declare integer *currentIndex*

declare string *result* and set to “”

while (true)

increment *numProbes*

set *currentIndex* to (*lowerBound* + *upperBound*)/2

if (*nameSearchKey* and *name* of *currentIndex* of myPresidents array are equal)

set *result* to “Found”

exit while loop

// end if

else // *nameSearchKey* and *name* in *currentIndex* of myPresidents array aren’t equal

if (*lowerBound* is greater than *upperBound* of myPresidents array)

set *result* to “Not Found”

exit while loop

// end if

else // *lowerBound* is less than or equal to *upperBound* of myPresidents array

if (*name* in *currentIndex* of myPresidents array is alphabetically before *nameSearchKey*)

set *lowerBound* to *currentIndex* + 1

// end if

else // *name* in *currentIndex* of myPresidents array is alphabetically after *nameSearchKey*

set *upperBound* to *currentIndex* – 1

// end else

// end else

// end else

// end while

print formatted *nameSearchKey*, *result*, and *numProbes*

// end binarySearch()

binarySearchDisplay()

print formatted title of binary search

print formatted header with column labels for Search Argument, Result, and Number of Probes

// end binarySearchDisplay()