**PresidentsMDrvr Class**

**PresidentsMDrvr(int max)**

Initialize new Presidents array called myPresidents of size 44

//end constructor

**add(Presidents president)**

for (each index in myPresidents array, set counter to first index, increment counter)

set current index of myPresidents to Presidents object passed as parameter

//end for

//end add()

**display(int dispType)**

switch (dispType)

case: unsorted

print formatted header

for (each index in myPresidents array, set counter to first index, increment counter)

print Presidents object in each index of myPresidents array to a formatted string

//end for

case: sorted

print formatted header

for (each index in myPresidents array, set counter to first index, increment counter)

print Presidents object in each index of myPresidents array to a formatted string

//end for

//end switch

//end display()

**bubbleSort(int sortType)**

declare out, counter for outer loop

declare in, counter for inner loop

if (sortType is sort by number)

for (each index in myPresidents array, set counter out to last index, decrement counter)

for (each index in myPresidents array, set counter in to first index, increment counter)

if (President’s number at index in is greater than President’s number at index in+1)

swap(in, in+1) values

//end if

//end inner loop

//end outer loop

//end if

if (sortType is sort by name)

for (each index in myPresidents array, set counter out to last index, decrement counter)

for (each index in myPresidents array, set counter in to first index, increment counter)

if (President’s name at index in is great than President’s name at index in+1)

swap(in, in+1) values

//end if

//end inner loop

//end outer loop

//end if

//end bubbleSort()

**swap(int pos1, int pos2)**

declare Presidents object named temp and set to index pos1 in myPresidents

set president object in index pos1 to index pos2

set president object in index pos2 to temp

//end swap()

**sequentialSearch(String[] inKey)**

declare int array hits of length 7

for (each index in inKey array, set counter a to first index, increment counter)

for (each index in myPresidents array, set counter i to first index, increment counter)

if (President’s party at index i equals search term at index a)

increment hits for search term at index a

//end if

//end for

//end for

displaySeqSearch(inKey, hits)

//end sequentialSearch

**displaySeqSearch(String[] inKey, int[] hits)**

print formatted header

for (each index in hits array, set counter a to first index, increment counter)

if (hits at index a is greater than 1)

print search argument, “Found” and number of hits

else

print search argument, “not found” and 0 hits

//end if

//end for

//end displaySeqSearch

**binarySearch(String[] binKey)**

declare int lowerBound, set to first index of myPresidents array

declare int upperBound, set to last index of myPresidents array

declare int curIn

declare int array binaryFound of length 6

declare int array probes of length 6

for (each index in binKey array, set counter i to first index, increment counter)

while (lowerBound is less than or equal to upperBound)

set curIn to (lowerBound + upperBound)/2

if (president’s name at curIn index of myPresidents array equals search term at index i of binKey array)

increment probes

increment binaryFound

break while

else if ( lowerBound is greater than upperBound, search term is not found)

break while

else

if (president’s name at curIn of myPresidents array is alphabetically before search term at index i of binKey array)

set lowerBound to curIn+1

increment probes

else(if president’s name at curIn of myPresidents array is alphabetically after search term at index i of binKey array)

set upperBound to curIn-1

increment probes

//end if

//end if

//end while

Set lowerBound to first index of myPresidents array

Set upperBound to last index of myPresidents array

//end for

displayBinsearch(binKey, probes, binaryFound)

//end binarySearch()

**displayBinSearch(String[] binKey, int[] probes, int[] binaryFound)**

print formatted header

for (each index in the binaryFound array, set counter a to first index, increment counter)

if (binaryFound is greater than 0)

print search argument, “found” and number of probes

else

print search argument, “not found” and number of probes

//end if

//end for

//end displayBinSearch

//end PresidentsMDrvr