Daniel Moe

Programming II, SD 254

July 25, 2012

Lab 2

Compiler: Dev-C++

Files: gcd.cpp

gcd.exe

|  |  |  |
| --- | --- | --- |
| Score | | |
|  | Team | Individual |
| Content & Quality |  |  |
| Format & mechanics |  |  |
| Timeliness |  |  |

**Lab 2 Pseudo Code**

**Linear Search**

Start

Int index

Int rtrn

Bool found

Array list[length]

rtrn = -1

WHILE (index < length AND !found)

If (liIst[index] == item)

Rtrn = index

Else

Increment index

END WHILE

If (found)

Return rtrn

**Main**

Int dataArray [ARRAYMAX]

Int choice

Int item

Int index

Int answer

Bool again

Bool sorted = false

Call function: initArray (dataArray, ARRAYMAX, DATAMAX)

Call function: printArray (dataArray, ARRAYMAX)

Loop

Prompt user choice: (1) sort or (2) search

Get choice

If (choice = sort)

If(list is unsorted)

call function sortArray(dataArray, ARRAYMAX)

call function printArray(dataArray, ARRAYMAX)

prompt user for search criteria

get criteria: item

index = binarysearch(item, dataArray, ARRAYMAX)

print index where item was found

else if(choice = search)

if(list is unsorted)

Prompt user for search criteria: int item

call function linearSearch(index, dataArray, ARRAYMAX)

print index where item was found

end if

end if

prompt user: search again? Yes/no

if answer == yes

again = true

if answer == no

again = false

End loop

**Lab 2 Source Code**

**Srchsort.cpp**

#include <cstdlib>

#include <iostream>

#include "arrays.h"

using namespace std;

int main(int argc, char \*argv[])

{

int dataArray [ARRAYMAX];

int choice;

int item;

int index;

int answer;

bool again;

bool sorted = false;

initArray (dataArray, ARRAYMAX, DATAMAX);

printArray (dataArray, ARRAYMAX);

do

{

cout << "Would you like to 1: sort the array or 2: search for a number? ";

cin >> choice;

if(choice == 1)

{

if(sorted == false)

{

sortArray (dataArray, ARRAYMAX);

printArray (dataArray, ARRAYMAX);

}

cout << endl << "What are you searching for? ";

cin >> item;

index = binsearch(item, dataArray, ARRAYMAX);

cout << endl << "Array[" << index << "]";

}

else if(choice == 2)

{

if(sorted == false)

{

cout << endl << "What are you searching for? ";

cin >> item;

index = linsearch(item, dataArray, ARRAYMAX);

cout << endl << "Array[" << index << "]";

}

}

cout << endl << "Would you like to search again? yes (1) /no (2): ";

cin >> answer;

if(answer == 1)

again = true;

if(answer == 2)

again = false;

}while(again == true);

system("PAUSE");

return EXIT\_SUCCESS;

}

**Array.cpp**

#include <cstdlib>

#include <iostream>

#include "arrays.h"

using namespace std;

void initArray (int A[], int arraySize, int modValue)

{

int i;

srand (time (NULL));

for (i=0; i<arraySize; i++)

A[i] = rand() % modValue;

}

void printArray (int A[], int arraySize)

{

int i;

for (i=0; i<arraySize; i++)

cout << "Array[" << i << "]" << " =" << A[i] << endl;

}

void sortArray (int A[], int arraySize)

{

int i;

int j;

int indexMin;

int valueMin;

for (i=0; i <ARRAYMAX -1; i++)

{

indexMin = i;

valueMin = A[i];

for (j=i+1; j <arraySize; j++)

{

if (A[j] < valueMin)

{

valueMin = A[j];

A[j] = A[i];

}

A[indexMin] = A[i];

A[i] = valueMin;

}

}

}

int linsearch(int item, int dataArray[], int ARRAYMAX)

{

int index = 0;

int rtrn;

bool found;

rtrn = -1;

while(index < ARRAYMAX && rtrn == -1)

{

if(dataArray[index] == item)

rtrn = index;

else

index++;

}

return rtrn;

}

int binsearch(int item, int dataArray[], int ARRAYMAX)

{

int left = 0;

int right = ARRAYMAX - 1;

int mid;

bool found = false;

while(!found && left <= right)

{

mid = (left + right)/2;

if(dataArray[mid] == item)

found = true;

else if(dataArray[mid] > item)

right = mid - 1;

else

left = mid + 1;

}

if(found)

return mid;

else

return -1;

}

**Array.h**

#include <cstdlib>

#include <iostream>

const int ARRAYMAX = 16;

const int DATAMAX = 127;

void initArray (int A[], int arraySize, int modValue);

void printArray (int A[], int arraySize);

void sortArray (int A[], int arraySize);

int linsearch(int item, int dataArray[], int ARRAYMAX);

int binsearch(int item, int dataArray[], int ARRAYMAX);

**Program Output**

