**CSC 1101 – Problem Solving and Programming Laboratory**

**Lab 20– Omar Faruk**

**25 points – Due Nov 24, end of class**

**a)** Save this document with your name and the homework number somewhere in the file name.

**b)** Type/paste your answers into the document.

**c)** Submit this document and your .cpp file(s) to the Canvas item where you downloaded this document. Do not submit a zip file but individually attach your files.

You are hired by Amazon to work on their product search team. Every product has a unique ID. Assume, product IDs are in sorted order. As a software engineer, you are required to make an amazon product search engine which works for every user.

Initialize an array size=7, and set the values {5, 12, 23, 27, 31, 52, 90} in your program. Then use a sentinel loop to prompt for and get from the user a series of IDs. For each ID, perform a binary search on the array and tell the user if it was found or not. If found, print the index where the ID is stored. Continue to prompt the user for an ID until entering the sentinel value of -999.

We know binary search divides the array by its half and search to the array. It finds the value when item== arr[mid];

For example, it will consider the full array for first search, for second search it will consider one of the half arrays and so on. You need to print that selected portion of the array for each search.

Inside the binary\_search function, you can do the followings to print that portion of the array:

For i=low to high

Print(array[i])

Low and high are defined in binary search usually.

**Guidelines:**

In addition to function **main**, define the following function to perform binary search:

**int binary\_search(int sort\_arr[], int size\_array, int search\_value)**

Take the array and search item from the main function and pass it to the **binary\_search** function.

**binary\_seacrh** returns the index of search value. If the value does not exist in the array, it returns -1. Moreover, this function prints the selection portion of the array for each search.

See **Arrays – linear and binary searches** sample application on Canvas.

Use the proper alignment according to the given input output.

**Sample I/O**

Text

Description automatically generated

*[your program code here]\**

//==========================================================

//

// Title: Amazon Product ID Locator

// Course: CSC 1101

// Lab Number: 20

// Author: Omar Faruk

// Date: 11/24/2020

// Description:

// Creating an app to locate product ID index

// from a list of predefinied product IDs using

// arrays, functions, and loops.

//

//==========================================================

#include <cstdlib> // For several general-purpose functions

#include <fstream> // For file handling

#include <iomanip> // For formatted output

#include <iostream> // For cin, cout, and system

#include <string> // For string data type

using namespace std; // So "std::cout" may be abbreviated to "cout"

// Gloabals

const int COLMFT1 = 20;

const int COLMFT2 = 15;

int binary\_search(int sort\_arr[], int size\_array, int search\_value)

{

// Declare variables

int min = 0;

int index;

int max = size\_array - 1;

// Loop to find key

while (min <= max)

{

// Get midpoint of array

index = (min + max) / 2;

// Test if key found

if (sort\_arr[index] == search\_value) // Guess is right on

return index;

else if (sort\_arr[index] < search\_value) // Guess too low

min = index + 1;

else

max = index - 1;

}

return -1;

}

void printArray(string heading, int arr[], int arrSize)

{

// Loop to print array numbers

cout << "\n" + heading << endl;

cout << "Array size: " << arrSize << endl;

cout << setw(COLMFT1) << left << "Index"

<< setw(COLMFT2) << left << "Value" << endl;

for (int i = 0; i < arrSize; i++)

cout << setw(COLMFT1) << left << i

<< setw(COLMFT1) << left << arr[i] << endl;

}

int main()

{

// Declare constants

const int ARRAYSIZE = 7;

// Declare variables

int ID\_index = 0;

int user\_input;

// Declare array

int UniqueID\_Array[ARRAYSIZE] = { 5, 12, 23, 27, 31, 52, 90 };

// Show application header

cout << "Welcome to Amazon Product ID Locator!" << endl;

cout << "--------------------------" << endl << endl;

printArray("Print the ID of each amazon product: ", UniqueID\_Array, ARRAYSIZE);

cout << "\nEnter a product ID to search in the array: (-999 to exit)" << endl;

cin >> user\_input;

while (user\_input != -999)

{

ID\_index = binary\_search(UniqueID\_Array, ARRAYSIZE, user\_input);

if (ID\_index == -1)

{

cout << "\nValue does not exist in the Uique ID list";

}

else

{

printArray("Search number 1: ", UniqueID\_Array, ARRAYSIZE);

cout << endl << user\_input << " is found in position: " << ID\_index << " of this sorted list";

}

cout << endl << "\nEnter a product ID to search in the array: (-999 to exit)" << endl;

cin >> user\_input;

}

// Show application close

cout << "\nEnd of Amazon Product ID Locator" << endl;

}

*[your program output here]\*\**



