Lab 09 Searching

Linear Search

int linearSearch( const int array[], int key, int size )

{

for ( int i = 0; i < size; i++ ) {

if ( array[i] == key ) {

return i; //return the location in array

}

}

return -1; //key not found

}

Binary Search

int binarySearch( const int array[], int key, int low, int high )

{

int middle; //the middle index number of the array

while ( low <= high ) {

middle = ( low + high ) / 2;

if ( key == array[ middle ] ) {

return middle;

}

else if ( key < array[ middle ] ) {

high = middle - 1; //reset high index to left side

}

else {

low = middle + 1; //reset low index to right side

}

}

return -1; //key not found

}

Question 1

Write a program to prompt the user to key in 5 elements into an array and a value they want to search from the array.

Use linearSearch and binarySearch to find the location of the key from the array. Display the location of the key in the array or a message “Key not found” if the key does not exist in the array.

-Steps to do:

Declare variables and array with 5 elements to store integer.

Prompt user to key in 5 integer numbers and store in your array.

Asked you user to key in a search key/ target.

Call you linear search to get the position of the target in the array.  
Call you binary search to get the position of the target in the array.

Display the position if you found the value or “Key not found” if the key not in the array.

Question 2

Suppose that the list L contains the integers 20, 60, 70, 100, 180, 190, 250, 280, 300 and 305. Trace through the steps of Sequential Search to determine what comparisons of keys are done in searching for the following targets:

(i) 190

(ii) 305

Question 3

Suppose that the list L contains the integers 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60 and 65. Trace through the steps of Binary Search to determine what comparisons of keys are done in searching for the following targets:

1. 60

**Step 1:**

Set approximate midpoint, index = (0+11)/2 = 5.5 = 6

Compare 60 and 40(midpoint), not matched,

{10,15,20,25,30,35,40,45,50,55,60,65}

60 > 40, go to right subarray

Lower limit of subarray = midpoint+1

**Step 2:**

Set approximate midpoint, index = (0+4)/2 = 2

Compare 60 and 55(midpoint), not matched,

{45,50,55,60,65}

60 > 55, go to right subarray

Lower limit of subarray = midpoint+1

**Step 3:**

Set approximate midpoint, index = (0+1)/2 = 0.5 = 1

Compare 60 and 65(midpoint), not matched,

{60,65}

60 < 65, go to left subarray

Upper limit of subarray = midpoint-1

**Step 4:**

Size of subarray = 1

Compare the element 60 and 60(only element in array), match is found

{60}

(iii) 30

**Step 1:**

Set approximate midpoint, index = (0+11)/2 = 5.5 = 6

Compare 30 and 40(midpoint), not matched,

{10,15,20,25,30,35,40,45,50,55,60,65}

30 < 40, go to left subarray

Upper limit of subarray = midpoint-1

**Step 2:**

Set approximate midpoint, index = (0+5)/2 = 2.5 = 3

Compare 30 and 25(midpoint), not matched,

{10,15,20,25,30,35}

30 > 25, go to right subarray

Lower limit of subarray = midpoint+1

**Step 3:**

Set approximate midpoint, index = (0+1)/2 = 0.5 = 1

Compare 30 and 35(midpoint), not matched,

{30,35}

30 < 35, go to left subarray

Upper limit of subarray = midpoint-1

**Step 4:**

Size of subarray = 1

Compare the element 30 and 30(only element in array), match is found

{30}