Binary Search

Theory

The linear search compares each element of the Linear Array 'A' which has N values in it with the search key until the search key is found.
To determine that a value is not in the array, the program must compare the search key to every element in the array 'A'. It is also called "Sequential Search" because it traverses the data sequentially to locate the element.

Suppose DATA is an array that is sorted in increasing (or decreasing) numerical order or, equivalently, alphabetically. Then there is an extremely efficient searching algorithm, called binary search,
which can be used to find the location LOC of a given ITEM of information in DATA.

Binary search on a list of elements stored in an array.

The binary search algorithm applied to our array DATA works as follows:

During each stage of algorithm, our search for ITEM is reduced to a segment of elements of DATA:

- DATA[BEG], DATA[BEG+1], DATA[BEG+2], . . ., DATA[END]

- Note that the variables BEG and END denote, respectively, the beginning and end locations of the segment under consideration. The algorithm compares ITEM with the middle element DATA[MID] of the segment, where MID is computed as

MID = INT((BEG + END) / 2) (INT(A) refers to the integer value of A.)

If DATA[MID] = = ITEM, then the search is successful and we set LOC = MID.
Otherwise a new segment of DATA is obtained as follows:

a) If ITEM < DATA[MID], then ITEM can appear only in the left half of the segment:
DATA[BEG], DATA[BEG+1], . . . , DATA[MID-1]
So we reset END = MID –1 and begin searching again.

b) If ITEM > DATA[MID], then ITEM can appear only in the right half of the segment:
DATA[MID+1], DATA[MID+2], . . . , DATA[END]
So we reset BEG = MID +1 and begin searching again.

Initially, we begin with the entire array DATA; i.e., we begin with

BEG = 0 and END = N-1.

If ITEM is not in DATA, then eventually we obtain BEG > END.

This condition signals that the search is unsuccessful, and in such a case we assign LOC = NULL. Here NULL is a value that lies outside the set of indices of DATA.

ALGORITHM

BINARY_SEARCH(DATA, ITEM)

1. Set BEG = 0, END = N-1 and MID = INT((BEG + END) / 2)
2. Repeat steps 3 and 4 while BEG <= END AND DATA[MID] ≠ ITEM
3. If ITEM < DATA[MID], then Set END = MID – 1, Else Set BEG = MID + 1
4. MID = INT((BEG + END) / 2)
5. If DATA[MID] = ITEM, then Set LOC = MID Else Set LOC = NULL
6. Exit

Lab Task:

1. Write a C program that makes use of the above algorithms as a function. Your program should input the array from the user.