Using the binary search algorithm.

Theory: Binarry Search is a searching algorithm used in a sorted array by repeatedly dividing the search interval in half. The idea of binarry search is the idea of binarry search is to use the information that the arrray is soruted and reduce the time complexity to O(log(n)).

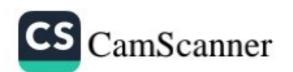
The binary search algorithm applied to our array DATA works as follows.

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

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

The algorithm compares ITEM with the the middle element DATA[MID] of the segment.

MID = INT ((BEG+ END)12)





If DATA [MID] = ITEM, then the search is successful and we set Loc = MID. Otherwise a new segment of DATA is Obtain 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

(b) If ITEM > DATA[MID] then ITEM can appear only in the right half of the segment:

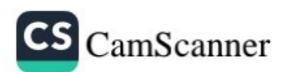
DATA [MID+1], DATA [MID+2], ... DATA (END) So we reset BEG = MID+1

Herre Grenerally the BGH BEG = LB (Lower Bound), End = (UB (Upper Bound).

Example:

DATA: 1,3,5,6,7,8,10,12

Suppose ITEM = 5. The searceh for ITEM in the array data.





\*\*Externe BEG = 1 and END = 8 Hence,

MID = INT ((1+8)/2) = 4 50 DATACMID] = 6

2. Since 5 < 6, END has its value changed

by END = MID - 1 = 4 - 2 = 3. Hence,

MID = INT ((1+3)/2) = 2 and DATACMID] = 3

3. Since 30 5>3, BEG has changed its

BEG = MID+1=82+1=3

MID = 1NT ((3+3)12) = 3, SO DATA [MID] = 5

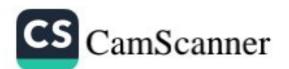
we have found ITEM in location LOC = MID = 5



Algorithm:
(Binary Search) BINARY (DATA, 1B, VB, STEM) of
Here DATA is sorted array which lower
bound LB, upper bound LB, Searching item
is item. Find location is Loc.

- 1. Set BEG = LB, FND = UB and MID=INT ((BEG+END)
- 2. Repeat step 3 and 4 while BEGGEND and DATAEMID] + ITEM.
- 3. If ITEM < DATAL MIDJ, then set END = MID-1
  else BEG = MID+1
- 4. set MID = INt (BEG+END)/2)
- 5. If DATACMIO] = ITEM, them, Set LOC = MID else set LOC = NULL

6. Exit.





```
Source code: 4+
 # include < iostream>
 using namespace std;
 int main()
    int DATA[10] = {1,2,3,4,5,6,7,8,9,10};
    int N=10, LB=0, UB=N-1, LOC;
    int BEG=LB, END=UB, MID=int((BEG+ENL
     int ITEM = 5;
     While (BEGZ=END & DATA [MID]!=ITEM)
        if (ITEM < DATA [MID])
           END=MID-1;
        else
BEG=MID+1;
   MID = int ((BEGTENO)12);

}
if (DATACMIO) = ITEM)
   else
Loc="NULL";
Cout 22" The searching element is: " LL DATA (LOC) 22 endl;
Cout 22" The searching element location is: "
```





Input:

Data elemets arre.

{ 1,2,3,4,5,6,7,8,9,10} Searcching element/ITEM = 5

output: ba

The serrachino

The searching element is: 5

The searcehing elements location is: 4

