

### Solution for problem 14:

Binary search is a basic divide and conquer technique that provides us the option for finding a specific value from a **sorted** set of values very quickly. Say you have a sorted array of integers, and you need to find a value  $v$  in it. The binary search approach will just divide the array into two halves, and then determine whether your value is on the left half or the right half. And if it is in the left/right half, then it will divide this half again.. and keep doing so unless it gets the position, or fails to find it (these are the base cases here). If you want to learn a bit in details, visit [here](#).

```
[.c]
#include <stdio.h>

int binarySearch(int i, int j, int v, int *a)
{
    int m;
    if(i > j) return -1;
    m = (i+j)/2;
    if(a[m] < v) return binarySearch(m+1, j, v, a);
    if(a[m] > v) return binarySearch(i, m-1, v, a);
    return m;
}

int main()
{
    int a[100], n, i, q, v, x;
    scanf("%d", &n);
    // give sorted input
    // otherwise sort the array after taking input
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);

    scanf("%d", &q);
    for(i=0; i<q; i++)
    {
        scanf("%d", &v);
        x = binarySearch(0, n-1, v, a);
        if(x==-1) printf("Not found\n");
        else printf("Found at %d\n", x);
    }
    return 0;
}
[.c]
```

\*Remember, binary search will only work for sorted array. Don't waste much time here if you

don't really want to learn for now, this will also be covered later... This is not very important for upcoming exam.