| 1 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

l

Key = 2

r

m

$m = 0+6/2 = 3$

$m' = 0+2/2 = 1$

$m'' = 0+0/2 = 0$

Stop

$l > r$

$l \leq r$

$2 < 3$

$2 > 1$

# Binary Search in an Array

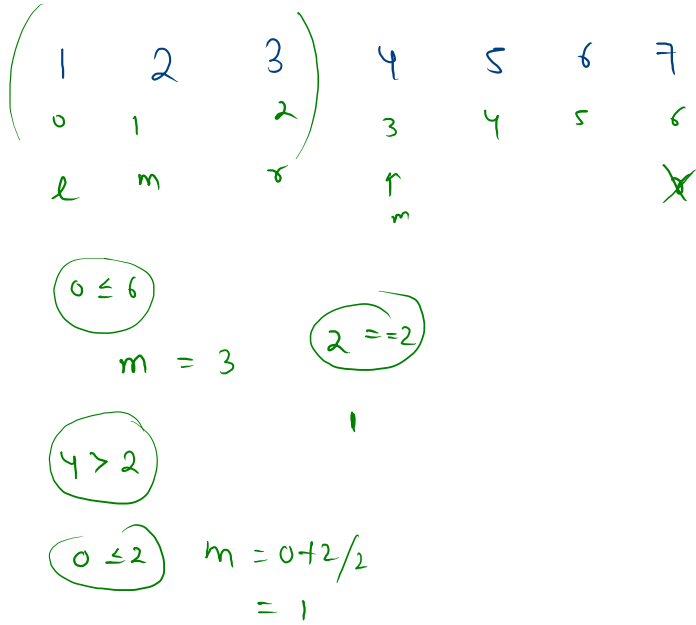
Problem | Submissions | Leaderboard | Discussions

Given a array of n elements sorted in the **increasing** order, write a function to search a given element **x** in array and print the **index** of x in the array.

**Note:** Consider the array is 0 based index and also that x definitely lies in range [0,arr.length].

key = 2

```
5   public static int binarySearch(int [] A, int x){
6       int left = 0;
7       int right = A.length-1;
8
9       while(left <= right){
0           int m = (left + right)/2;
1           if(A[m] == x){
2               return m;
3           }
4           else if(A[m] > x){
15              right = m-1;
16          }
17          else{
18              left = m + 1;
19          }
20      }
21      return -1;
22  }
```



```java
import java.io.*;
import java.util.*;

public class Solution {
    public static int binarySearch(int [] A, int x){
        int left = 0;
        int right = A.length-1;

        while(left <= right){
            int m = (left + right)/2;
            if(A[m] == x){
                return m;
            }
            else if(A[m] > x){
                right = m-1;
            }
            else{
                left = m + 1;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int [] A = new int[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.nextInt();
        }
        int x = scn.nextInt();
        int ans = binarySearch(A, x);
        System.out.println(ans);

    }
}
```

# Search Character

Problem | Submissions | Leaderboard | Discussions

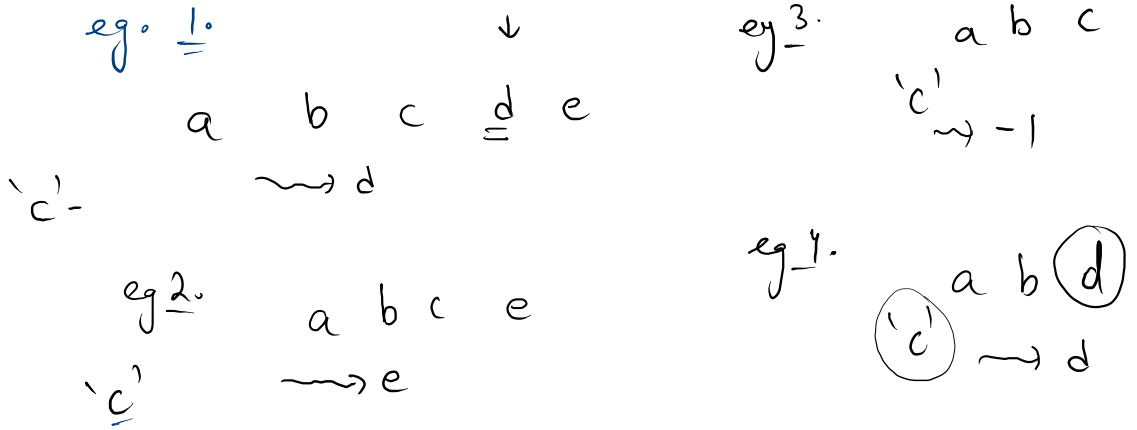
Given a small case character ch and a sorted array containing only the small case alphabets, you have to print the alphabet just greater than the character ch present in array. If no such character found print -1.

**Sample Input 0**

```
c
5
a b c d e
```

**Sample Output 0**

```
d
```




d

e

f

g

h

i

j

k

...

z

a b c f g

0 1 2 3 4

idx = 2

f [ 2+1 ]

a b c

0 1 2

idx = 2

-1

ch = 'c'

a b c f g

a b c

a b f

```java
import java.io.*;
import java.util.*;

public class Solution {
    public static int binarySearch(char [] A, char key){
        int left = 0;
        int right = A.length-1;
        while(left <= right){
            int m = (left + right)/2;
            if(A[m] == key){
                return m;
            }
            else if(A[m] > key){
                right = m-1;
            }
            else{
                left = m+1;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        char key = scn.next().charAt(0);
        int n = scn.nextInt();
        char [] A = new char[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.next().charAt(0);
        }
        //next potential ans
        key++;
        for(char c = key; c <= 'z' ; c++){
            int idx = binarySearch(A, c);
            if(idx != -1){
                System.out.println(A[idx]);
                return;
            }
        }
        System.out.println(-1);
    }
}
```

```java
import java.io.*;
import java.util.*;

public class Solution {
    public static int binarySearch(char [] A, char key){
        int left = 0;
        int right = A.length-1;
        while(left <= right){
            int m = (left + right)/2;
            if(A[m] == key){
                return m;
            }
            else if(A[m] > key){
                right = m-1;
            }
            else{
                left = m+1;
            }
        }
        return -1;
    }
```

```java

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        char key = scn.next().charAt(0);
        int n = scn.nextInt();
        char [] A = new char[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.next().charAt(0);
        }
        //next potential ans
        key++;
        for(char c = key; c <= 'z' ; c++){
            int idx = binarySearch(A, c);
            if(idx != -1){
                System.out.println(A[idx]);
                return;
            }
        }
        System.out.println(-1);
    }
}
```

key = 'c'

a b g

ans =

```java
import java.io.*;
import java.util.*;

public class Solution {
    public static int binarySearch(char [] A, char key){
        int left = 0;
        int right = A.length-1;
        while(left <= right){
            int m = (left + right)/2;
            if(A[m] == key){
                return m;
            }
            else if(A[m] > key){
                right = m-1;
            }
            else{
                left = m+1;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        char key = scn.next().charAt(0);
        int n = scn.nextInt();
        char [] A = new char[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.next().charAt(0);
        }
        //next potential ans
        key++;
        for(char c = key; c <= 'z' ; c++){
            int idx = binarySearch(A, c);
            if(idx != -1){
                System.out.println(A[idx]);
                return;
            }
        }
        System.out.println(-1);
    }
}
```

Last Occurence.

$x = 3$ { }

ans = ~~x~~ 4

Sample Input 0

```
6
1 2 3 3 3 4
3
```

| 1 | 2 | 3 | 3 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 |
| | | | ~~3~~ | = | r |
| | | | | | l |

Sample Output 0

```
4
```

$m = 0+5/2 = 2$ $3+5/2 = 4$

$A[m] == x$

$3 == 3$

ans = m

$l = m + 1$

Key = 3

| 1 | 2 | 3 | 3 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | (4) | 5 | 6 | 7 | 8 |
| | | | | r | l | | | |

l <= r

ans = 4

5 > 3

4 > 3

$m = 0 + 8/2 = 4$

$m_1 = 5 + 8/2 = 6$

$m_2 = 5 + 5/2 = 5$

```java
public class Solution {
    public static int binarySearch(int [] A, int x){
        int left = 0;
        int right = A.length-1;
        int ans = -1;
        while(left <= right){
            int m = (left + right)/2;
            if(A[m] == x){
                ans = m;
                //last occ
                left = m + 1;
            }
            else if(A[m] > x){
                right = m-1;
            }
            else{
                left = m + 1;
            }
        }
        return ans;
    }
```

key = 3

| 1 | 2 | 3 | 3 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 |

ans = -1 2 4

r

l

0 ≤ 5 ✓ m = 0+5/2 = 2

A[2] == 3

3 == 3

5 ≤ 4

5 ≤ 5 ✓ m = 5+5/2 = 5

4 > 3

A[5] == 3

4 == 3 ✗

3 ≤ 5 m = 3+5/2 = 4

A[4] == 3

3 == 3