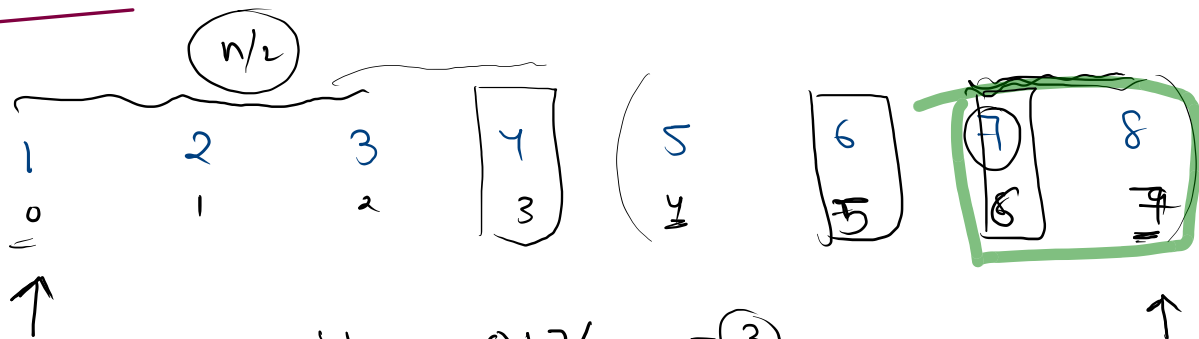


Binary Search

$$\underline{n=7}$$



$$\text{mid} = 0 + 7/2 = 3$$

3 checks

$$A[\text{mid}] = A[3] = k$$

$$\text{mid} = 4 + 7/2 = 11/2 = 5$$

$$\text{mid} = 6 + 7/2 = 13/2 = 6$$

$k=2$

1	2	3	4	5	6	7	8
0	①	2	3	4	5	6	7
		high					
low							

2.

$$\underline{\text{mid} = 1}$$

$$\text{mid} = \text{low} + \text{high} / 2$$

$$\text{mid} = 0 + 7 / 2 = 3$$

if ($A[\text{mid}] == k$)
 return mid; ✓

else if ($A[\text{mid}] > k$)
 {
 high = mid - 1
 }

eg. 3.

1		h		l				
1	2	3	<u>4</u>	5	6	7	<u>8</u>	
0	1	2	3	4	5	6	7	
l							high, low	

⇒ K = 16

mid = (7) ✓

low ≤ high

K > A[mid]
correct range

high < low

↳ incorrect range.

```

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 key = scn.nextInt(); // 5
    int ans = -1;

    int low = 0;
    int high = n-1;

    while(low <= high){
        int mid = (low + high)/2;

        if(key == A[mid]){
            ans = mid;
            break;
        }
        else if(key > A[mid]){
            // dir -> right
            low = mid + 1;
        }
        else{
            // key < A[mid] : dir -> left
            high = mid - 1;
        }
    }

    System.out.println(ans);
}

```

1 2 3
0 1 2

4 5 6 7 8
3 4 5 6 7
h m

k=5

ans = -1
n = 8
low = 4
high = 7

$l \leq h$
 $4 \leq 4$

mid = 4

ans = 4

$A[4] = 5 == 5$

it's

1 \textcircled{n}
2 $\frac{n}{2}$
3 $\frac{n/2}{2} = \frac{n}{4}$
4 $\frac{n/4}{2} = \frac{n}{8}$
last 1

$$\Rightarrow \overbrace{\left[n \quad n/2 \quad \textcircled{\frac{n}{4}} \quad \frac{n}{8} \quad \dots \right]}^{\textcircled{k}} 1$$

$$\Rightarrow \frac{n}{2^0} \quad \frac{n}{2^1} \quad \frac{n}{2^2} \quad \frac{n}{2^3} \quad \dots \quad \frac{n}{2^k}$$

$$\textcircled{1 = \frac{n}{2^k}}$$

$$n = 2^k$$

$$\log_2 n = k \log_2 2 \rightarrow 1$$

$$\boxed{k = \log_2 n}$$

$$O(1) \quad \textcircled{\log n} \quad n \quad n \log n \quad n^2 \quad n^3$$

$$\text{int } \underline{\underline{\text{mid}}} = \frac{\overset{\text{Int. Max}}{\underline{\underline{\text{low}}}} + \overset{\text{Intca.}}{\underline{\underline{\text{high}}}}}{2} \quad \} \quad \checkmark$$

69. Sqrt(x)

Easy 6758 4147 Add to List Share

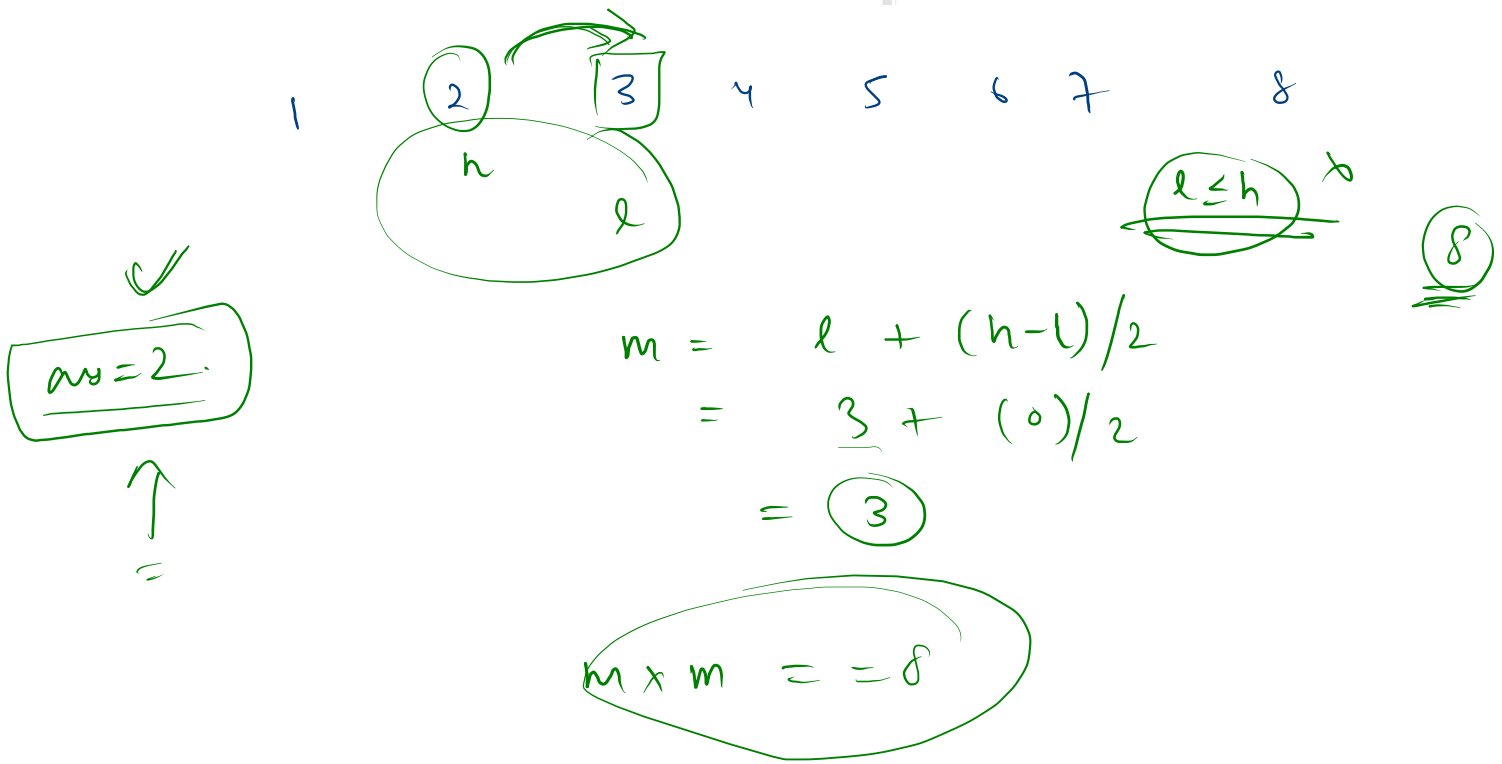
Given a non-negative integer x , return the square root of x rounded down to the nearest integer. The returned integer should be non-negative as well.

You must not use any built-in exponent function or operator.

- For example, do not use `pow(x, 0.5)` in c++ or `x ** 0.5` in python.

$n = 8$
 $ans = 0$
 $low = 1$
 $high = n$

16



$$m * m = n$$

$$m = \frac{n}{m}$$

$$m * m = n$$

$$m = \frac{n}{m}$$

$$n = 16$$

$$as = 0$$

$$low = 1$$

$$high = \cancel{16} 7$$

$$m = 1 + (n - 1) / 2$$

$$= 1 + 6 / 2$$

$$\underline{4} \times 4 = \textcircled{= 16} = 4$$

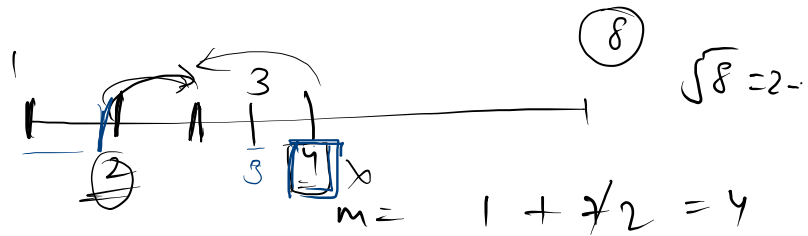
|
 e

))
 nm

```

1 class Solution {
2     public int mySqrt(int x) {
3         int ans = 0;
4         int low = 1;
5         int high = x;
6         while(low <= high){
7             int mid = low + (high - low)/2;
8
9             if(mid == x / mid){
10                 return mid;
11             }
12             else if(mid < x/mid){
13                 //right
14                 ans = mid;
15                 low = mid + 1;
16             }
17             else{
18                 //left ✓
19                 high = mid - 1;
20             }
21         }
22
23         return ans;
24     }
25 }

```



$$m = 1 + 2/2 = 2$$

$$4 \times 4 = 8$$

$$2 \times 2 \leq 8$$



Search Character

key = b → -1

n = 5

a	b	c	d	e
0	1	2	3	4

→ Binary Search. _{ch.}

a	b	<u>c</u>	f	g
		2		

d
r
t

key == c
f

key == c.

a	b	c	f	g	h
0	1	<u>2</u>	3	4	5

1. check key is
present or
not.
==

key == c

a	c	c	c
---	---	---	---

↓
expected

↑
Actual.

1. if \rightarrow ch = 'd'

You have to search
from e to z
which is occurring first
that will be
answer.

else
-1

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5     public static int search(char [] A, char key){
6         int low = 0;
7         int high = A.length-1;
8
9         while(low <= high){
10             int mid = low + (high-low)/2;
11
12             if(A[mid] == key){
13                 return mid;
14             }
15             else if(A[mid] > key){
16                 high = mid - 1;
17             }
18             else{
19                 low = mid + 1;
20             }
21         }
22         return -1;
23     }
24 }

```

```

23     }
24
25     public static void main(String[] args) {
26         Scanner scn = new Scanner(System.in);
27         char key = scn.next().charAt(0);
28         int n = scn.nextInt();
29         char [] A = new char[n];
30         for(int i = 0; i < n; i++){
31             A[i] = scn.next().charAt(0);
32         }
33
34         key++; //c will become d
35         for(char ch = key ; ch <= 'z'; ch++ ){
36             int ans = search(A,ch);
37             if(ans != -1){
38                 System.out.println(ch);
39                 return;
40             }
41         }
42         System.out.println(-1);
43     }

```