

Search Character

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
```

ch = 'e' → c
d
e
f

a b f k l
0 1 2 3 4

a d e f → ch = 'b'

ch ≥ 'a' and ch ≤ 'z'

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5     public static int binarySearch(char [] A, char x){
6         int i = 0;
7         int j = A.length-1;
8         while(i <= j){
9             int m = (i + j) / 2;
10            if(A[m] == x){
11                return m;
12            }
13            else if(A[m] > x){ //left
14                j = m - 1;
15            }else{           //right
16                i = m + 1;
17            }
18        }
19        return -1;
20    }

```

a c f g k

x = 'b'
n = 'c'

```

20
21 public static void main(String[] args) {
22     Scanner scn = new Scanner(System.in);
23     char x = scn.next().charAt(0); // 'b'
24     int n = scn.nextInt();
25     char [] A = new char[n];
26     for(int i = 0; i < n; i++){
27         A[i] = scn.next().charAt(0);
28     }
29
30     // b -> 'c'
31     x++;
32     for(char ch = x; ch <= 'z'; ch++){
33         int ans = binarySearch(A, ch);
34         if(ans != -1){
35             System.out.println(A[ans]);
36             return;
37         }
38     }
39     System.out.println(-1);
40 }
41 }

```

Find Last Occurrence

$$x = 3$$

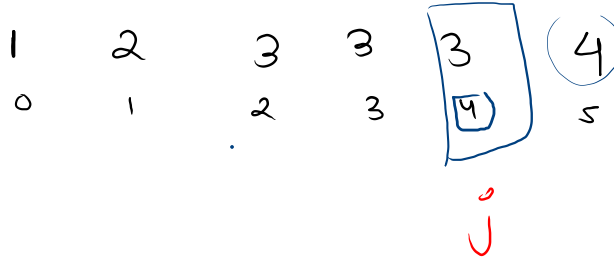
$$\text{ans} = \cancel{2} 4$$

Sample Input 0

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

Sample Output 0

```
4
```



$$A[m] == x$$

↳ ans = m

i
m

$$A[m] > x$$

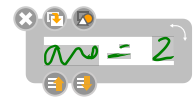
LO of 1'

$x = 1$

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

j

i m



$A[m] == x$ $ans = m$

$A[m] > x$

$$\begin{array}{cccccc}
 1 & 0 & 0 & 0 & 0 & 0 \\
 & \downarrow & & & & /2 \\
 5 & 0 & 0 & 0 & 0 & \\
 & \downarrow & & & & \\
 2 & 5 & 0 & 0 & 0 & \\
 & \downarrow & & & & \\
 1 & 2 & 5 & 0 & 0 &
 \end{array}$$

1

```

4 public class Solution {
5     public static int lastOcc(int [] A, int x){
6         int ans = -1;
7         int i = 0;
8         int j = A.length-1;
9         while(i <= j){
10             int m = (i + j) / 2;
11             if(A[m] == x){
12                 ans = m;
13                 i = m+1;
14             }else if(A[m] > x){ //left
15                 j = m-1;
16             }else{ //A[m] < x right
17                 i = m+1;
18             }
19         }
20         return ans;
21     }
22
23     public static void main(String[] args) {
24         Scanner scn = new Scanner(System.in);
25         int n = scn.nextInt();
26         int [] A = new int[n];
27         for(int i = 0; i < n; i++){
28             A[i] = scn.nextInt();
29         }
30         int x = scn.nextInt();
31         int ans = lastOcc(A, x);
32         System.out.println(ans);
33     }
34 }

```

$m = -1 \times 6$

$x = 2$

1	1	1	1	2	2	2	3	3
0	1	2	3	4	5	6	7	8
						j	i	
								m

$A[m] == x$

$3 > 2$

```
public static void main(String
    int i = 2147483647;    //2
    int j = 2147483647;    //2
    int m = (i+j)/2;
```

$$m = i + \frac{(j-i)}{2}$$

$$= i + \frac{j-i}{2}$$

$$= \frac{i}{2} + \frac{j}{2} = \frac{i+j}{2}$$

$$0/2 = 0$$

$$i + 0$$

2147483647

$$i = 0$$

$$j = 8$$

$$m = \frac{0+8}{2} = 4$$

$$0 + \frac{8}{2} = 0 + 4 = 4$$

Square Root

$$m + m = x$$

69. Sqrt(x)

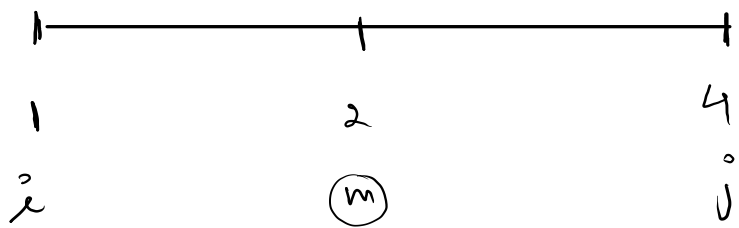
Easy 8276 4518 Add to List Share

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.

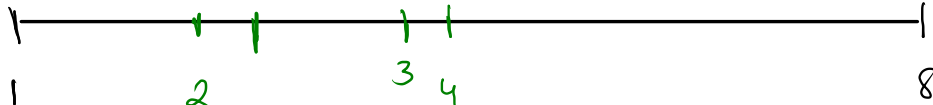
You **must not use** any built-in exponent function or operator.

$$x = 4$$

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



$$\omega = \phi 2$$



$$x=8 \rightarrow (2,3)$$

$$x=8$$

2.

$$m * m = x$$

$$m * m > x$$

$$m * m < x$$

$$\omega = m$$

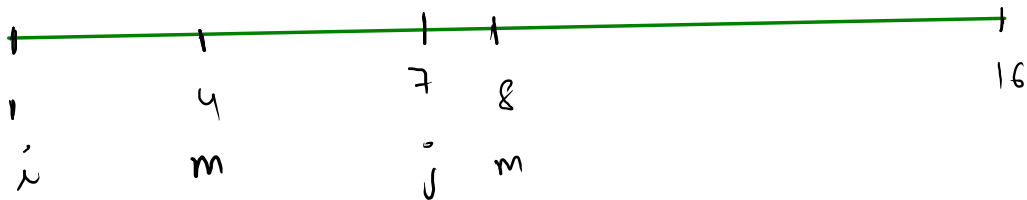
$$3 \times 3 = 8$$

2. -

$$(3)$$

$$x = 16$$

$$ms = \emptyset \quad 4$$



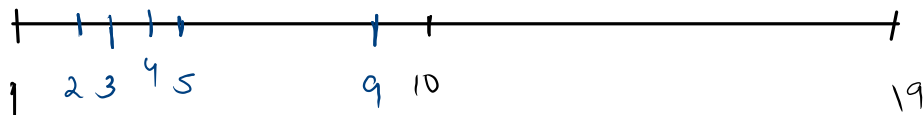
$$m * m == x$$

$$m * m > x$$

$$8 * 8 == x$$

$$aw = 6 \text{ of } 3 \text{ } 4$$

$$x = 19$$



$\overset{\circ}{j} \overset{\circ}{i}$

m

$$10 \times 10 = 19$$

$$5 \times 5 \Rightarrow 19$$

$$\textcircled{2 \times 2 \not\leq 19}$$

$$3 \times 3 < 19$$

$$4 \times 4 < 19$$

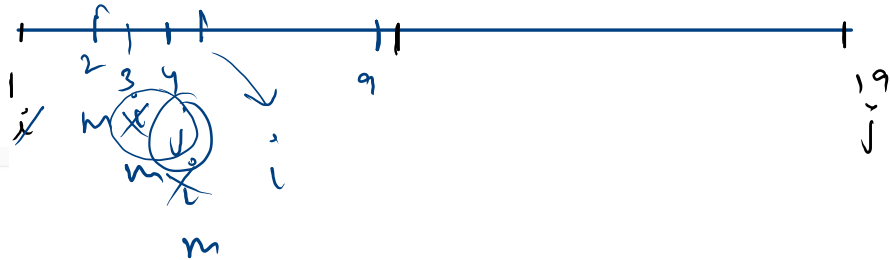
$$m \times m = x \rightarrow m = x/m$$

$$m \times m > x \quad m > x/m$$

$$m \times m < x \quad m < x/m$$

$$x = 19$$

$$ans = 4$$



$$5 \times 5 = 19$$

$$3 \times 3 < 19$$

$$4 \times 4 < 19$$

$$2 \times 2 = 19$$

```

1  class Solution {
2      public int mySqrt(int x) {
3          int i = 1;
4          int j = x;
5          int ans = 0;
6
7          while(i <= j){
8              int m = i + (j-i)/2;
9
10             if(m == x/m){
11                 ans = m; //return m;
12                 return ans;
13             }else if( m > x /m){ //m*m > x
14                 j = m - 1;
15             }else{ //m*m < x
16                 ans = m;
17                 i = m + 1;
18             }
19
20         }
21         return ans;
22     }
23 }
24

```