

**146.** Given an array `arr` of positive integers sorted in a strictly increasing order, and an integer `k`. return the `k`th positive integer that is missing from this array.

Program:-

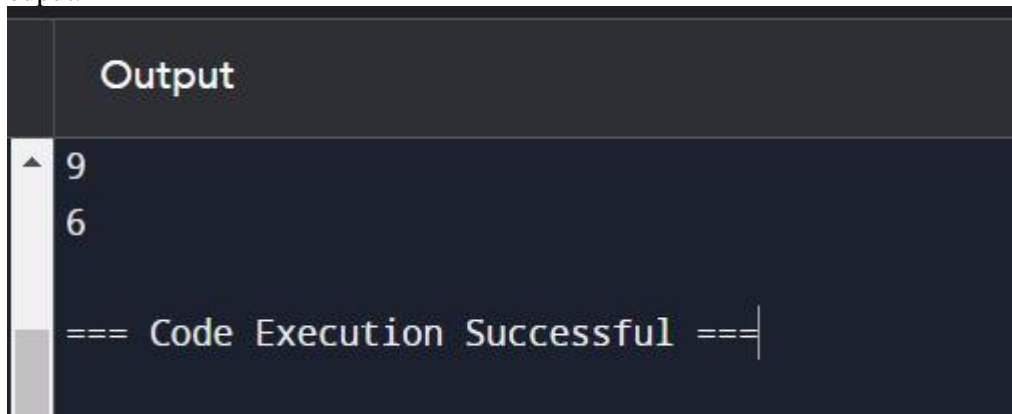
```
def find_kth_missing(arr, k):  
    i = 0  
    missing_count = 0  
  
    while i < len(arr):  
        if i == 0:  
            missing_in_between = arr[i] - 1  
        else:  
            missing_in_between = arr[i] - arr[i-1] - 1  
  
        if missing_count + missing_in_between >= k:  
            # kth missing integer is between arr[i-1] and arr[i]  
            return arr[i-1] + (k - missing_count)  
  
        missing_count += missing_in_between  
        i += 1
```

input:-

`arr1 = [2, 3, 4, 7, 11]`

`k1 = 5`

output:-

A screenshot of a code execution environment with a dark background. At the top, the word "Output" is displayed in a light blue font. Below it, the numbers "9" and "6" are shown in a light blue font, one above the other. At the bottom, the text "=== Code Execution Successful ===" is displayed in a light blue font, with a vertical cursor line at the end.

**Time complexity:-** $O(\log n)$