GeeksforGeeks
A computer science portal for geeks
Practice IDE Q&A GeeksQuiz

Print missing elements that lie in range 0 – 99

Given an array of integers print the missing elements that lie in range 0-99. If there are more than one missing, collate them, otherwise just print the number.

Note that the input array may not be sorted and may contain numbers outside the range [0-99], but only this range is to be considered for printing missing elements.

Examples

```
Input: {88, 105, 3, 2, 200, 0, 10}
Output: 1
        4-9
        11-87
        89-99


Input: {9, 6, 900, 850, 5, 90, 100, 99}
Output: 0-4
        7-8
        10-89
        91-98
```

Expected time complexity O(n), where n is the size of the input array.

We strongly recommend to minimize your browser and try this yourself first.

The idea is to use a boolean array of size 100 to keep track of array elements that lie in range 0 to 99. We first traverse input array and mark such present elements in the boolean array. Once all present elements are marked, the boolean array is used to print missing elements.

Following is C implementation of above idea.

```c
// C program for print missing elements
#include<stdio.h>
#define LIMIT 100

// A O(n) function to print missing elements in an array
void printMissing(int arr[], int n)
{
    // Initialize all number from 0 to 99 as NOT seen
    bool seen[LIMIT] = {false};

    // Mark present elements in range [0-99] as seen
    for (int i=0; i<n; i++)
      if (arr[i] < LIMIT)
       seen[arr[i]] = true;

    // Print missing element
    int i = 0;
    while (i < LIMIT)
    {
        // If i is missing
        if (seen[i] == false)
        {
            // Find if there are more missing elements after i
            int j = i+1;
            while (j < LIMIT && seen[j] == false)
                  j++;

            // Print missing single or range
            (i+1 == j)? printf("%d\n", i): printf("%d-%d\n", i, j-1);

            // Update u
            i = j;
        }
        else
            i++;
    }
}

// Driver program
int main()
{
    int arr[] = {88, 105, 3, 2, 200, 0, 10};
    int n = sizeof(arr)/sizeof(arr[0]);
    printMissing(arr, n);
    return 0;
}
```

Run on IDE

Output:

```
1
4-9
11-87
89-99
```

Time complexity of the above program is O(n).

This article is contributed by Vignesh Narayanan and Sowmya Sampath. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

27 Comments Category: Arrays

Related Posts:

- Longest Span with same Sum in two Binary arrays
- Count Inversions of size three in a give array
- Find the subarray with least average
- Count triplets with sum smaller than a given value
- Find zeroes to be flipped so that number of consecutive 1's is maximized
- Reorder an array according to given indexes
- Find maximum value of Sum(i*arr[i]) with only rotations on given array allowed
- Find maximum average subarray of k length

(Login to Rate and Mark)

2.5 Average Difficulty : **2.5/5.0**
Based on **7** vote(s)

□ Add to TODO List
□ Mark as DONE

Like Share 17 people like this. Be the first of your friends.

Writing code in comment? Please use code.geeksforgeeks.org, generate link and share the link here.

Contact Us! About Us! Advertise with us!