

# 【华为OD机考 统一考试机试C卷】小明找位置 (C++ Java JavaScript Python C语言)

## 华为OD机考:统一考试 C卷 + D卷 + B卷 +A卷

目前在考C卷，经过两个月的收集整理，**C卷真题已基本整理完毕**

抽到原题的概率为2/3到3/3，**也就是最少抽到两道原题。请注意：大家刷完C卷真题，最好要把B卷的真题刷一下，因为C卷的部分真题来自B卷。**

另外订阅专栏还可以联系笔者开通在线 **OJ** 进行刷题，提高刷题效率。

**真题目录：**华为OD机考机试 真题目录 (C卷 + D卷 + B卷 + A卷) + 考点说明

**专栏：**2023华为OD机试( B卷+C卷+D卷) (C++JavaJSPy)

**华为OD面试真题精选：**华为OD面试真题精选

**在线OJ：**点击立即刷题，模拟真实机考环境

## 题目描述

小朋友出操，按学号从小到大排成一列;小明来迟了，请你给小明出个主意，让他尽快找到他应该排的位置。

算法复杂度要求不高于 $n\log(n)$ ;学号为整数类型，队列规模 $\leq 10000$ ;

## 输入描述

1、第一行:输入已排成队列的小朋友的学号 (正整数)，以","隔开

例如: 93,95,97,100,102,123,155

2、第二行:小明学号，如110;

## 输出描述

输出一个数字，代表队列位置 (从1开始)例如:

6

## 用例

---

输入	93,95,97,100,102,123,155 110
输出	6
说明	无

## C++

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  int main() {
6      std::string line;
7      std::getline(std::cin, line); // 读取一行输入
8      int xiaoMingNumber;
9      std::cin >> xiaoMingNumber;
10
11     // 分割字符串并转换为整数数组
12     std::vector<int> numbers;
13     size_t pos = 0;
14     while ((pos = line.find(',')) != std::string::npos) {
15         numbers.push_back(stoi(line.substr(0, pos)));
16         line.erase(0, pos + 1);
17     }
18     numbers.push_back(stoi(line)); // 添加最后一个元素
19
20     // 对数组进行排序
21     std::sort(numbers.begin(), numbers.end());
22
23     // 使用二分查找找到小明的位置
24     auto position = std::lower_bound(numbers.begin(), numbers.end(), xiaoMingNumber) - numbers.begin();
25
26     // 输出小明应该排的位置, 位置从1开始计数
27     std::cout << position + 1 << std::endl;
28
29     return 0;
30 }
```

## Java

```
1  import java.util.*;
2
3  public class Main {
4      public static void main(String[] args) {
5
6          Scanner sc = new Scanner(System.in);
7          String line = sc.nextLine();
8          int xiaoMingNumber = Integer.parseInt(sc.nextLine());
9
10         // 将字符串分割成数组并转换为整数
11         String[] numbersStr = line.split(",");
12         int[] numbers = new int[numbersStr.length];
13         for (int i = 0; i < numbers.length; i++) {
14             numbers[i] = Integer.parseInt(numbersStr[i].trim());
15         }
16
17         // 对数组进行排序
18         Arrays.sort(numbers);
19
20         // 使用二分查找找到小明的位置
21         int position = Arrays.binarySearch(numbers, xiaoMingNumber);
22         // binarySearch()方法的返回值为:
23         // 1、如果找到关键字, 则返回值为关键字在数组中的位置索引, 且索引从0开始
24         // 2、如果没有找到关键字, 返回值为负的插入点值, 所谓插入点值就是第一个比关键字大的元素在数组中的位置索引,
25         // 而且这个位置索引从1开始。
26         // 如果位置是负数, 转换为插入位置
27         if (position < 0) {
28             position = -position - 1;
29         }
30
31         // 输出小明应该排的位置, 位置从1开始计数
32         System.out.println(position + 1);
33     }
34 }
```

## JavaScript

```
1  const readline = require('readline');
2
3  // 创建 readline 接口实例
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout
7  });
8
9
10 rl.on('line', (line) => {
11   // 读取小明的学号
12   rl.on('line', (xiaoMingNumber) => {
13     xiaoMingNumber = parseInt(xiaoMingNumber);
14
15     // 分割字符串并转换为整数数组
16     const numbers = line.split(',').map(num => parseInt(num.trim()));
17
18     // 对数组进行排序
19     numbers.sort((a, b) => a - b);
20
21     // 使用二分查找算法找到小明的位置
22     let left = 0, right = numbers.length, mid;
23     while (left < right) {
24       mid = Math.floor((left + right) / 2);
25       if (numbers[mid] < xiaoMingNumber) {
26         left = mid + 1;
27       } else {
28         right = mid;
29       }
30     }
31     const position = left;
32
33     // 输出小明应该排的位置, 位置从1开始计数
34     console.log(position + 1);
35     process.exit();
36   });
37 });
```

```

1 # 读取输入
2 line = input()
3 xiao_ming_number = int(input())
4
5 # 分割字符串并转换为整数列表
6 numbers = [int(num) for num in line.split(',')]
7
8 # 对列表进行排序
9 numbers.sort()
10
11 # 使用二分查找找到小明的位置
12 position = 0
13 left, right = 0, len(numbers)
14 while left < right:
15     mid = (left + right) // 2
16     if numbers[mid] < xiao_ming_number:
17         left = mid + 1
18     else:
19         right = mid
20 position = left
21
22 # 输出小明应该排的位置, 位置从1开始计数
23 print(position + 1)

```

## C语言

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 // 字符串分割函数
6 void split(const char *str, int *array, int *size) {
7     char *token;
8     char *strCopy = strdup(str); // 复制字符串, 避免修改原字符串
9
10    token = strtok(strCopy, ",");
11    while (token != NULL) {
12        array[*size] = atoi(token); // 将字符串转换为整数, 并存储在数组中
13        (*size)++;
14        token = strtok(NULL, ",");
15    }
16 }

```

```
15     }
16
17     free(strCopy); // 释放复制的字符串
18 }
19
20 // 比较函数, 用于qsort
21 int compare(const void *a, const void *b) {
22     return (*(int *)a - *(int *)b);
23 }
24
25 // 二分查找函数
26 int binarySearch(int *array, int size, int xiaoMingNumber) {
27     int low = 0, high = size - 1;
28     while (low <= high) {
29         int mid = low + (high - low) / 2;
30         if (array[mid] == xiaoMingNumber) {
31             return mid + 1; // 返回位置时加1, 因为位置从1开始计数
32         }
33         if (array[mid] < xiaoMingNumber) {
34             low = mid + 1;
35         } else {
36             high = mid - 1;
37         }
38     }
39     return low + 1; // 如果找不到, 返回小明应插入的位置
40 }
41
42 int main() {
43     char line[10000]; // 存储输入的字符串
44     fgets(line, 10000, stdin); // 读取一行输入
45     int xiaoMingNumber;
46     scanf("%d", &xiaoMingNumber);
47
48     int numbers[10000], size = 0;
49     split(line, numbers, &size); // 分割字符串并转换为整数数组
50
51     qsort(numbers, size, sizeof(int), compare); // 对数组进行排序
52
53     // 使用二分查找找到小明的位置
54     int position = binarySearch(numbers, size, xiaoMingNumber);
55 }
```

```
56 |  
57 | // 输出小明应该排的位置  
58 | printf("%d\n", position);  
59 |  
60 | return 0;  
   | }  
   |
```

## 完整用例

### 用例1

93,95,97,100,102,123,155  
110

### 用例2

93,95,97,100,102,123,155  
90

### 用例3

93,95,97,100,102,123,155  
160

### 用例4

93,95,97,100,102,123,155  
123

### 用例5

100  
110

### 用例6

100  
90

用例7

93,95,97,100,102,123,155  
93

用例8

93,95,97,100,102,123,155  
155

用例9

93,95,97,100,100,102,123,155  
110

用例10

93,94,95,95,96,97,98,99,100  
95

文章目录

- 华为OD机考:统一考试 C卷 + D卷 + B卷 +A卷
  - 题目描述
  - 输入描述
  - 输出描述
  - 用例
  - C++
  - Java
  - javaScript
  - Python
  - C语言
  - 完整用例
    - 用例1
    - 用例2
    - 用例3
    - 用例4
    - 用例5



用例6  
用例7  
用例8  
用例9  
用例10

# 机考真题 华为OD



CSDN @算法大师