【华为OD机考 统一考试机试C卷】按身高和体重排队（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

某学校举行运动会，学生们按编号(1、2、3…n)进行标识，现需要按照身高由低到高排列，
对身高相同的人，按体重由轻到重排列；
对于身高体重都相同的人，维持原有的编号顺序关系。请输出排列后的学生编号。

输入描述

两个序列，每个序列由n个正整数组成（0 < n <= 100）。

第一个序列中的数值代表身高，第二个序列中的数值代表体重。

输出描述

排列结果，每个数值都是原始序列中的学生编号，编号从1开始

用例1

输入

```
4
100 100 120 130
40 30 60 50
```

输出

```
2 1 3 4
```

说明

> 输出的第一个数字2表示此人原始编号为2，即身高为100，体重为30的这个人。
>
> 由于他和编号为1的人身高一样，但体重更轻，因此要排在1前面。

用例2

输入

```
3
90 110 90
45 60 45
```

输出

```
1 3 2
```

说明

> 1和3的身高体重都相同，需要按照原有位置关系让1排在3前面，而不是3 1 2。

C++

```
#include <iostream>
#include <vector>

```

```cpp
#include <algorithm>
using namespace std;

int main() {
    // 定义变量
    int n;
    vector<int> heights, weights;
    // 输入人数
    cin >> n;
    // 调整向量大小
    heights.resize(n);
    weights.resize(n);
    // 输入身高
    for (int i = 0; i < n; i++) {
        cin >> heights[i];
    }
    // 输入体重
    for (int i = 0; i < n; i++) {
        cin >> weights[i];
    }
    // 定义人的向量
    vector<int> persons(n);
    // 初始化人的编号
    for (int i = 0; i < n; i++) {
        persons[i] = i + 1;
    }
    // 按照身高和体重排序
    sort(persons.begin(), persons.end(), [&](int a, int b) {
        if (heights[a-1] == heights[b-1]) {
            return weights[a-1] < weights[b-1];
        } else {
            return heights[a-1] < heights[b-1];
        }
    });
    // 输出人的编号
    for (int i = 0; i < n; i++) {
        cout << persons[i] << " ";
    }
    cout << endl;


```

```
    return 0;
}
```

javaScript

```javascript
const readline = require('readline');
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

rl.on('line', (input) => {
  const n = parseInt(input);
  const heights = new Array(n);
  const weights = new Array(n);
  const persons = new Array(n).fill(0).map((_, i) => i + 1);

  rl.on('line', (input) => {
    const arr = input.split(' ').map(Number);
    for (let i = 0; i < n; i++) {
      if (heights[i] === undefined) heights[i] = arr[i];
      else weights[i] = arr[i];
    }

    if (weights[n - 1] !== undefined) {
      persons.sort((a, b) => {
        if (heights[a - 1] === heights[b - 1]) {
          return weights[a - 1] - weights[b - 1];
        } else {
          return heights[a - 1] - heights[b - 1];
        }
      });

      console.log(persons.join(' '));
      rl.close();
    }
  });
});
```

java

```java
import java.util.Scanner;
import java.util.Vector;
import java.util.Collections;

public class Main {
    public static void main(String[] args) {
        // 定义变量
        int n;
        Vector<Integer> heights = new Vector<Integer>();
        Vector<Integer> weights = new Vector<Integer>();
        // 输入人数
        Scanner scanner = new Scanner(System.in);
        n = scanner.nextInt();
        // 调整向量大小
        heights.setSize(n);
        weights.setSize(n);
        // 输入身高
        for (int i = 0; i < n; i++) {
            heights.set(i, scanner.nextInt());
        }
        // 输入体重
        for (int i = 0; i < n; i++) {
            weights.set(i, scanner.nextInt());
        }
        // 定义人的向量
        Vector<Integer> persons = new Vector<Integer>();
        // 初始化人的编号
        for (int i = 0; i < n; i++) {
            persons.add(i + 1);
        }
        // 按照身高和体重排序
        Collections.sort(persons, (a, b) -> {
            if (heights.get(a-1).equals(heights.get(b-1))) {
                return weights.get(a-1).compareTo(weights.get(b-1));
            } else {
                return heights.get(a-1).compareTo(heights.get(b-1));
            }
        });

```

```
        // 输出人的编号
        for (int i = 0; i < n; i++) {
            System.out.print(persons.get(i) + " ");
        }
        System.out.println();
    }
}
```

python

```
import sys

n = int(sys.stdin.readline().strip())
heights = [0] * n
weights = [0] * n
persons = list(range(1, n+1))

def handle_input(input):
    arr = list(map(int, input.strip().split()))
    for i in range(n):
        if heights[i] == 0:
            heights[i] = arr[i]
        else:
            weights[i] = arr[i]

    if weights[n-1] != 0:
        persons.sort(key=lambda x: (heights[x-1], weights[x-1]))
        print(' '.join(map(str, persons)))
        sys.exit()

while --n:
    input_str = sys.stdin.readline()
    handle_input(input_str)
```

C语言

```
#include <stdio.h>
#include <stdlib.h>

```

```c

#define MAX_STUDENTS 100

// 定义学生结构体
typedef struct {
    int id; // 学生编号
    int height; // 学生身高
    int weight; // 学生体重
} Student;

// 定义全局变量
int n; // 学生数量
Student students[MAX_STUDENTS]; // 存储学生对象的数组

// 定义比较函数，用于 qsort 函数
int cmp(const void *a, const void *b) {
    Student *studentA = (Student *)a;
    Student *studentB = (Student *)b;
    if (studentA->height != studentB->height) {
        return studentA->height - studentB->height; // 身高低的排前面
    }
    if (studentA->weight != studentB->weight) {
        return studentA->weight - studentB->weight; // 体重轻的排前面
    }
    return studentA->id - studentB->id; // 身高体重都相同则按编号顺序排列
}

int main() {
    scanf("%d", &n);
    for (int i = 0; i < n; ++i) {
        students[i].id = i + 1;
        scanf("%d", &students[i].height);
    }
    for (int i = 0; i < n; ++i) {
        scanf("%d", &students[i].weight);
    }

    // 对学生进行排序
    qsort(students, n, sizeof(Student), cmp);


```

```
    // 输出排序后的学生编号
    for (int i = 0; i < n; ++i) {
        printf("%d ", students[i].id);
    }
    printf("\n");

    return 0;
}
```

完整用例

用例1

4
100 100 120 130
40 30 60 50

用例2

3
90 110 90
45 60 45

用例3

5
150 160 170 180 190
50 60 70 80 90

用例4

6
200 200 200 200 200 200
70 80 90 100 110 120

用例5

2
140 140
50 60

用例6

10
180 170 160 150 140 130 120 110 100 90
80 70 60 50 40 30 20 10 5 3

用例7

7
150 160 170 180 190 200 210
50 60 70 80 90 100 110

用例8

8
100 110 120 130 140 150 160 170
80 70 60 50 40 30 20 10

用例9

9
100 100 100 100 100 100 100 100 100
90 80 70 60 50 40 30 20 10

用例10

5
80 90 100 110 120
20 30 40 50 60

文章目录

机考真题
华为OD