

【华为OD机考 统一考试机试C卷】考勤信息 (C++ Java JavaScript Python)

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

公司用一个字符串来表示员工的出勤信息

- absent: 缺勤
- late: 迟到
- leaveearly: 早退
- present: 正常上班

现需根据员工出勤信息，判断本次是否能获得出勤奖，能获得出勤奖的条件如下：

- 缺勤不超过一次；
- 没有连续的迟到/早退；
- 任意连续7次考勤，缺勤/迟到/早退不超过3次。

输入描述

用户的考勤数据字符串

- 记录条数 >= 1;
- 输入字符串长度 < 10000;
- 不存在非法输入;

如：

```
2
present
present absent present present leaveearly present absent
```

输出描述

根据考勤数据字符串，如果能得到考勤奖，输出"true"；否则输出"false"，对于输入示例的结果应为：

```
true false
```

用例

输入	2 present present present
输出	true true
说明	无

输入	2 present present absent present present leaveearly present absent
输出	true

输入	2
	present present absent present present leaveearly present absent
	false
说明	无

C++

```

1  #include <iostream>
2  #include <sstream>
3  #include <vector>
4
5  using namespace std;
6  // 检查考勤记录是否符合获得考勤奖的条件
7  bool check(vector<string> records) {
8      int absent = 0; // 记录缺勤次数
9      // 遍历考勤记录
10     for (int i = 0; i < records.size(); i++) {
11         // 如果当前记录是缺勤
12         if (records[i] == "absent") {
13             absent++; // 缺勤次数增加
14             // 如果缺勤超过一次, 则不满足条件
15             if (absent > 1) {
16                 return false;
17             }
18         }
19         // 如果当前记录是迟到或早退
20         else if (records[i] == "late" || records[i] == "leaveearly") {
21             // 如果前一条记录也是迟到或早退, 则不满足条件
22             if (i > 0 && (records[i - 1] == "late" || records[i - 1] == "leaveearly")) {
23                 return false;
24             }
25         }
26     }
27     // 检查任意连续7次考勤
28     if (i >= 6) {
29         int count = 0; // 记录连续7次考勤中非正常上班的次数
30         // 遍历连续的7次记录
31         for (int j = i - 6; j <= i; j++) {

```

```

32         // 如果记录不是正常上班, 则计数增加
33         if (records[j] != "present") {
34             count++;
35         }
36     }
37     // 如果非正常上班的次数超过3次, 则不满足条件
38     if (count > 3) {
39         return false;
40     }
41 }
42 }
43 // 如果所有条件都满足, 则返回true
44 return true;
45 }
46
47 int main() {
48     int n;
49     cin >> n;
50     cin.ignore();
51
52     for (int i = 0; i < n; i++) {
53         string line;
54         getline(cin, line);
55         istringstream iss(line);
56         vector<string> records;
57         for (string s; iss >> s; ) {
58             records.push_back(s);
59         }
60         cout << (check(records) ? "true" : "false") << endl;
61     }
62
63     return 0;
64 }

```

java

```

1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5

```

```

5 // 创建Scanner对象用于读取输入
6 Scanner scanner = new Scanner(System.in);
7 // 读取第一行输入, 表示记录条数
8 int n = scanner.nextInt();
9 // 读取换行符
10 scanner.nextLine();
11
12 // 循环读取每一条记录
13 for (int i = 0; i < n; i++) {
14     // 读取一行记录并按空格分割, 存入字符串数组
15     String[] records = scanner.nextLine().split(" ");
16     // 调用check方法检查是否能得到考勤奖, 并输出结果
17     System.out.println(check(records) ? "true" : "false");
18 }
19 }
20
21 // 检查考勤记录是否符合获得考勤奖的条件
22 public static boolean check(String[] records) {
23     int absent = 0; // 记录缺勤次数
24
25     // 遍历考勤记录
26     for (int i = 0; i < records.length; i++) {
27         // 如果当前记录是缺勤
28         if (records[i].equals("absent")) {
29             absent++; // 缺勤次数增加
30             // 如果缺勤超过一次, 则不满足条件
31             if (absent > 1) {
32                 return false;
33             }
34         }
35         // 如果当前记录是迟到或早退
36         else if (records[i].equals("late") || records[i].equals("leaveearly")) {
37             // 如果前一条记录也是迟到或早退, 则不满足条件
38             if (i > 0 && (records[i - 1].equals("late") || records[i - 1].equals("leaveearly"))) {
39                 return false;
40             }
41         }
42     }
43     // 检查任意连续7次考勤
44     if (i >= 6) {
45         int count = 0; // 记录连续7次考勤中非正常上班的次数

```

```

46         // 遍历连续的7次记录
47         for (int j = i - 6; j <= i; j++) {
48             // 如果记录不是正常上班, 则计数增加
49             if (!records[j].equals("present")) {
50                 count++;
51             }
52         }
53         // 如果非正常上班的次数超过3次, 则不满足条件
54         if (count > 3) {
55             return false;
56         }
57     }
58 }
59 // 如果所有条件都满足, 则返回true
60 return true;
61 }
}

```

javaScript

```

1  const readline = require('readline').createInterface({
2      input: process.stdin,
3      output: process.stdout
4  });
5
6  // 检查考勤记录是否符合获得考勤奖的条件
7  function check(records) {
8      let absent = 0; // 记录缺勤次数
9      // 遍历考勤记录
10     for (let i = 0; i < records.length; i++) {
11         // 如果当前记录是缺勤
12         if (records[i] === "absent") {
13             absent++; // 缺勤次数增加
14             // 如果缺勤超过一次, 则不满足条件
15             if (absent > 1) {
16                 return false;
17             }
18         }
19         // 如果当前记录是迟到或早退
20         else if (records[i] === "late" || records[i] === "leaveearly") {
21

```

```

21 // 如果前一条记录也是迟到或早退, 则不满足条件
22 if (i > 0 && (records[i - 1] === "late" || records[i - 1] === "leaveearly")) {
23     return false;
24 }
25 }
26 // 检查任意连续7次考勤
27 if (i >= 6) {
28     let count = 0; // 记录连续7次考勤中非正常上班的次数
29     // 遍历连续的7次记录
30     for (let j = i - 6; j <= i; j++) {
31         // 如果记录不是正常上班, 则计数增加
32         if (records[j] !== "present") {
33             count++;
34         }
35     }
36     // 如果非正常上班的次数超过3次, 则不满足条件
37     if (count > 3) {
38         return false;
39     }
40 }
41 }
42 // 如果所有条件都满足, 则返回true
43 return true;
44 }
45
46 let lineCount = 0;
47 let n = 0;
48
49 readline.on('line', (line) => {
50     if (lineCount === 0) {
51         n = parseInt(line);
52     } else {
53         const records = line.split(' ');
54         console.log(check(records) ? "true" : "false");
55         if (lineCount === n) {
56             readline.close();
57         }
58     }
59     lineCount++;
60 });

```

python

```
1 # 检查考勤记录是否符合获得考勤奖的条件
2 def check(records):
3     absent = 0 # 记录缺勤次数
4     # 遍历考勤记录
5     for i in range(len(records)):
6         # 如果当前记录是缺勤
7         if records[i] == "absent":
8             absent += 1 # 缺勤次数增加
9             # 如果缺勤超过一次, 则不满足条件
10            if absent > 1:
11                return False
12        # 如果当前记录是迟到或早退
13        elif records[i] == "late" or records[i] == "leaveearly":
14            # 如果前一条记录也是迟到或早退, 则不满足条件
15            if i > 0 and (records[i - 1] == "late" or records[i - 1] == "leaveearly"):
16                return False
17        # 检查任意连续7次考勤
18        if i >= 6:
19            count = 0 # 记录连续7次考勤中非正常上班的次数
20            # 遍历连续的7次记录
21            for j in range(i - 6, i + 1):
22                # 如果记录不是正常上班, 则计数增加
23                if records[j] != "present":
24                    count += 1
25            # 如果非正常上班的次数超过3次, 则不满足条件
26            if count > 3:
27                return False
28        # 如果所有条件都满足, 则返回true
29        return True
30
31 n = int(input())
32 for _ in range(n):
33     records = input().split()
34     print("true" if check(records) else "false")
```

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

机考C卷真题

华为OD

