

【华为OD机考 统一考试机试C卷】 出租车计费 、 靠谱的车 (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：出租车计费 、 靠谱的车

程序员小明打了一辆出租车去上班。出于职业敏感，他注意到这辆出租车的计费表有点问题，总是偏大。

出租车司机解释说他不喜欢数字4，所以改装了计费表，任何数字位置遇到数字4就直接跳过，其余功能都正常。

比如：

1. 23再多一块钱就变为25；
2. 39再多一块钱变为50；
3. 399再多一块钱变为500；

小明识破了司机的伎俩，准备利用自己的学识打败司机的阴谋。

给出计费表的表面读数，返回实际产生的费用。

输入描述

只有一行，数字N，表示里程表的读数。

(1<=N<=888888888)。

输出描述

一个数字，表示实际产生的费用。以回车结束。

用例

输入	5
输出	4
说明	5表示计费表的表面读数。4表示实际产生的费用其实只有4块钱。

输入	17
输出	15
说明	17表示计费表的表面读数。15表示实际产生的费用其实只有15块钱。

输入	100
输出	81
说明	100表示计费表的表面读数。81表示实际产生的费用其实只有81块钱。

解题思路

我们需要理解出租车司机改装计费表的方式。司机不喜欢数字4，所以他的计费表从1开始计数，直到3，然后跳过4，继续从5计数。这意味着计费表上的每一位数字实际上只有9种可能的值：0, 1, 2, 3, 5, 6, 7, 8, 9（跳过了4）。因此，这个计费表实际上是在使用一个基数为9的数制系统。

现在，让我们用三个用例来解释这个过程：

- 1. 输入：5
输出：4

解释：由于计费表跳过了数字4，所以表面读数5实际上是9进制中的5。在10进制中，这是4。

2. 输入：17

输出：15

解释：表面读数17在9进制中表示为 $1 * 9^1 + 7 * 9^0$ 。但是，因为我们跳过了4，所以实际的计算应该是 $1 * 9^1 + 6 * 9^0$ （因为7实际上是8，但我们要减去1以补偿跳过的4），这等于 $9 + 6 = 15$ 。

3. 输入：100

输出：81

解释：表面读数100在9进制中表示为 $1 * 9^2 + 0 * 9^1 + 0 * 9^0$ 。在10进制中，这等于81。

因此，通过将表面读数视为9进制数，并将其转换为10进制数，同时考虑到跳过的4，我们可以得到实际产生的费用。这就是为什么我们使用9进制进行求解的原因。

特殊9进制数	10进制数	9进制展开形式
1	1	$1 * 9^0$
2	2	$2 * 9^0$
3	3	$3 * 9^0$
5	4	$4 * 9^0$
6	5	$5 * 9^0$
7	6	$6 * 9^0$
8	7	$7 * 9^0$
9	8	$8 * 9^0$
10	9	$19^1 + 09^0$
11	10	$19^1 + 19^0$
12	11	$19^1 + 29^0$
13	12	$19^1 + 39^0$
15	13	$19^1 + 49^0$

特殊9进制数	10进制数	9进制展开形式
16	14	$19^1 + 59^0$
17	15	$19^1 + 69^0$
18	16	$19^1 + 79^0$
19	17	$19^1 + 89^0$
20	18	$29^1 + 09^0$
21	19	$29^1 + 19^0$
22	20	$29^1 + 29^0$

在这个特殊的9进制系统中，我们跳过了数字4，所以当我们看到特殊9进制数15时，实际上是指真实9进制数13，它对应于10进制中的13（ $19^1 + 49^0$ ）。注意，在这个系统中，我们将特殊9进制数转换为真实9进制数时，需要将每个大于4的数字减去1来得到真实的9进制数，然后再将其转换为10进制数。

C++

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main() {
6      string line;
7      // 读取输入的表面读数
8      getline(cin, line);
9      // 初始化实际产生的费用
10     int correct = 0;
11     // 遍历读数的每一位数字
12     for (char c : line) {
13         int digit = c - '0'; // 将字符转换为数字
14         if (digit > 4) { // 如果数字大于4，则需要减1
15             digit--;
16         }
17         correct = correct * 9 + digit; // 将每一位数字加入到实际产生的费用中
18     }
19 }
```

```
20 // 输出实际产生的费用
21 cout << correct << endl;
22
23 return 0;
24 }
```

java

```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         String line = scanner.nextLine();
7         int correct = 0;
8         for (char c : line.toCharArray()) {
9             int digit = c - '0';
10            if (digit > 4) {
11                digit--;
12            }
13            correct = correct * 9 + digit;
14        }
15        System.out.println(correct);
16    }
17 }
```

javascript

```
1 const readline = require('readline');
2
3 const rl = readline.createInterface({
4     input: process.stdin,
5     output: process.stdout
6 });
7
8 rl.on('line', (line) => {
9     let correct = 0;
10    for (let i = 0; i < line.length; i++) {
11
```

```

12     let digit = parseInt(line[i]);
13     if (digit > 4) {
14         digit--;
15     }
16     correct = correct * 9 + digit;
17 }
18 console.log(correct);
19 });

```

python

```

1 import sys
2
3 line = sys.stdin.readline().strip()
4 correct = 0
5 for c in line:
6     digit = int(c)
7     if digit > 4:
8         digit -= 1
9     correct = correct * 9 + digit
10 print(correct)
11

```

C语言

```

1 #include <stdio.h>
2 #include <string.h>
3
4 int main() {
5     char line[10];
6     scanf("%s", line); // 读取输入的表面读数
7
8     int correct = 0; // 初始化实际产生的费用
9     int length = strlen(line); // 获取输入字符串的长度
10
11     // 遍历读数的每一位数字
12     for (int i = 0; i < length; i++) {
13         int digit = line[i] - '0'; // 将字符转换为数字
14

```

```
15
16     // 如果数字大于4，则需要减1，因为跳过了数字4
17     if (digit > 4) {
18         digit--;
19     }
20
21     correct = correct * 9 + digit; // 更新实际产生的费用
22 }
23
24 // 输出实际产生的费用
25 printf("%d\n", correct);
26
27 return 0;
}
```

完整用例

用例1

5

用例2

17

用例3

162

用例4

323

用例5

500

用例6

888888888

用例7

987321

用例8

12367

用例9

12367

用例10

777777

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: 出租车计费 、 靠谱的车

输入描述

输出描述

用例

解题思路

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