

【华为OD机考 统一考试机试C卷】求字符串中所有整数的最小和 (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

输入字符串s，输出s中包含所有整数的最小和。

说明：

字符串s，只包含 a-z A-Z ± ；

合法的整数包括

- 正整数 一个或者多个0-9组成，如 0 2 3 002 102
- 负整数 负号 (-) 开头，数字部分由一个或者多个0-9组成，如 -0 -012 -23 -00023

输入描述

包含数字的字符串

输出描述

所有整数的最小和

用例1

输入

```
1 | bb1234aa
```

输出

```
1 | 10
```

用例2

输入

```
1 | bb12-34aa
```

输出

```
1 | -31
```

说明

```
1+2+ (-34) = -31
```

C++

```
1 | #include <iostream>
2 | #include <string>
3 | #include <regex>
4 |
5 | using namespace std;
6 |
7 | // 计算整数各个位数之和
8 | int getNumSum(string num) {
9 |     int sum = 0;
10 |    for (char c : num) {
11 |        sum += c - '0';
12 |    }
```

```

12     }
13     return sum;
14 }
15
16 int main() {
17     string line;
18     getline(cin, line);
19     // 用正则表达式分割出所有整数
20     regex reg("[^\\d-]+");
21     sregex_token_iterator it(line.begin(), line.end(), reg, -1);
22     sregex_token_iterator end;
23     int res = 0;
24     for (; it != end; ++it) {
25         string num = *it;
26         if (!num.empty()) {
27             if (num.find("-") == string::npos) {
28                 // 处理正整数
29                 res += getNumSum(num);
30             } else {
31                 // 处理负整数
32                 bool isNeg = num[0] == '-';
33                 regex subReg("-");
34                 sregex_token_iterator subIt(num.begin(), num.end(), subReg, -1);
35                 int i = 0;
36                 for (; subIt != end; ++subIt) {
37                     string subNum = *subIt;
38                     if (!subNum.empty()) {
39                         int ele = stoi(subNum);
40                         if (i == 0) {
41                             res += isNeg ? 0 - ele : getNumSum(subNum);
42                         } else {
43                             res -= ele;
44                         }
45                         ++i;
46                     }
47                 }
48             }
49         }
50     }
51     cout << res << endl;
52 }

```

```
53 |
54 |     return 0;
    | }
```

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          // 用正则表达式分割出所有整数
8          String[] numArr = line.split("[^\\d-]+");
9          int res = 0;
10         for (String num : numArr) {
11             if (!num.isEmpty()) {
12                 if (num.indexOf("-") == -1) {
13                     // 处理正整数
14                     res += getNumSum(num);
15                 } else {
16                     // 处理负整数
17                     boolean isNeg = num.startsWith("-");
18                     String[] subNumArr = num.split("-");
19                     for (int i = 0; i < subNumArr.length; i++) {
20                         if (!subNumArr[i].isEmpty()) {
21                             int ele = Integer.parseInt(subNumArr[i]);
22                             if (i == 0) {
23                                 res += isNeg ? 0 - ele : getNumSum(subNumArr[i]);
24                             } else {
25                                 res -= ele;
26                             }
27                         }
28                     }
29                 }
30             }
31         }
32         System.out.println(res);
33     }
34 }
```

```

35
36 // 计算整数各个位数之和
37 private static int getNumSum(String num) {
38     int sum = 0;
39     for (char c : num.toCharArray()) {
40         sum += Character.getNumericValue(c);
41     }
42     return sum;
43 }
44 }

```

javaScript

```

1
2 const readline = require('readline');
3
4 const rl = readline.createInterface({
5     input: process.stdin,
6     output: process.stdout
7 });
8
9 rl.on('line', (line) => {
10     const numArr = line.split(/[^\d-]+/);
11     let res = 0;
12     for (const num of numArr) {
13         if (num) {
14             if (num.indexOf('-') === -1) {
15                 res += getNumSum(num);
16             } else {
17                 const isNeg = num.startsWith('-');
18                 const subNumArr = num.split('-');
19                 for (let i = 0; i < subNumArr.length; i++) {
20                     if (subNumArr[i]) {
21                         const ele = parseInt(subNumArr[i]);
22                         if (i === 0) {
23                             res += isNeg ? 0 - ele : getNumSum(subNumArr[i]);
24                         } else {
25                             res -= ele;
26                         }
27                     }
28                 }
29             }
30         }
31     }
32 }
33 }

```

```

28         }
29     }
30 }
31 }
32 console.log(res);
33 });
34
35 function getNumSum(num) {
36     let sum = 0;
37     for (const c of num) {
38         sum += parseInt(c);
39     }
40     return sum;
41 }

```

python

```

1  import re
2
3  def getNumSum(num):
4      sum = 0
5      for c in num:
6          sum += int(c)
7      return sum
8
9
10 line = input()
11 numArr = re.findall(r'[-?]\d+', line)
12 res = 0
13 for num in numArr:
14     if num:
15         if '-' not in num:
16             res += getNumSum(num)
17         else:
18             isNeg = num.startswith('-')
19             subNumArr = num.split('-')
20             for i in range(len(subNumArr)):
21                 if subNumArr[i]:
22                     ele = int(subNumArr[i])
23                     if i == 0:
24

```

```
24         res += -ele if isNeg else getNumSum(subNumArr[i])
25     else:
26         res -= ele
27     print(res)
```

完整用例

用例1

bb1234aa

用例2

bb12-34aa

用例3

a1b2c3d4

用例4

abc-12-34def

用例5

a0b0c0d0

用例6

a1b2c3d4e5f6g7h8i9j10k11

用例7

a-1b-2c-3d-4e-5f-6g-7h-8i-9j-10k-11

用例8

aaa123aaa456aaa789

用例9

a1b2c3-4d5e6f7-8g9h10i11

用例10

a-1b-2c-3d-4e-5f-6g-7h-8i-9j-10k11

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1

用例2

C++

java

javaScript

python

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