

【华为OD机考 统一考试机试C卷】小明的幸运数 (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

小明在玩一个游戏，游戏规则如下：在游戏开始前，小明站在坐标轴原点处（坐标值为0）。

给定一组指令和一个幸运数，每个指令都是一个整数，小明按照指令前进指定步数或者后退指定步数。前进代表朝坐标轴的正方向走，后退代表朝坐标轴的负方向走。

幸运数为一个整数，如果某个指令正好和幸运数相等，则小明行进步数+1。

例如：

幸运数为3，指令为[2,3,0,-5]

指令为2，表示前进2步；

指令为3，正好和幸运数相等，前进3+1=4步；

指令为0，表示原地不动，既不前进，也不后退。

指令为-5，表示后退5步。

请你计算小明在整个游戏过程中，小明所处的最大坐标值。

输入描述

第一行输入1个数字，代表指令的总个数 n ($1 \leq n \leq 100$)

第二行输入1个数字，代表幸运数 m ($-100 \leq m \leq 100$)

第三行输入 n 个指令，每个指令的取值范围为： $-100 \leq \text{指令值} \leq 100$

输出描述

输出在整个游戏过程中，小明所处的最大坐标值。异常情况下输出：12345

用例1

输入

1		2
2		1
3		-5 1

输出

1		0
---	--	---

说明

总共2个指令，幸运数为1，按照指令行进，依次如下游戏开始前，站在坐标轴原点，此时坐标值为0；
指令为-5，后退5步，此时坐标值为-5；
指令为1，正好等于幸运数，前进1+1=2步，此时坐标值为-3；
整个游戏过程中，小明所处的坐标值依次为[0, -5, -3]，最大坐标值为0。

用例2

输入

```
1 | 5
2 | -5
3 | -5 1 6 0 -7
```

输出

```
1 | 1
```

说明

总共5个指令，幸运数为-5，依照指令行进，依次如下：
游戏开始前，站在坐标轴原点，此时坐标值为0，
指令为-5，正好等于幸运数，后退 $5+1=6$ 步，此时坐标值为-6；
指令为1，前进1步，此时坐标值为-5；
指令为6，前进6步，此时坐标值为1；
指令为0，既不前进，也不后退，此时坐标值为1；
指令为-7，后退7步，此时坐标值为-6。
整个游戏过程中，小明所处的坐标值依次为：
[0, -6, -5, 1, 1, -6]，最大坐标值为1。

解题思路

本题就模拟计算,主要注意幸运数字是0的情况.

C++

```
1 | #include <iostream>
2 | #include <vector>
3 | #include <sstream>
4 | #include <algorithm>
5 |
6 | using namespace std;
7 |
8 | int main() {
9 |     // 创建一个新的扫描器实例
10 |     string line;
11 |
12 |
```

```
13 // 读取用户输入的整数n和l
14 getline(cin, line);
15 int n = stoi(line);
16 getline(cin, line);
17 int lucky = stoi(line);
18
19 // 检查输入的整数n和l是否在指定的范围内
20 if (n < 1 || n > 100 || lucky < -100 || lucky > 100) {
21     cout << "12345" << endl;
22     return 0;
23 }
24
25 // 初始化位置变量p和最大位置变量mp
26 int p = 0;
27 int mp = 0;
28
29 // 读取用户输入的整数cmd
30 getline(cin, line);
31 stringstream ss(line);
32 vector<int> cmds;
33 for (int i = 0; i < n; i++) {
34     int cmd;
35     ss >> cmd;
36     cmds.push_back(cmd);
37 }
38
39 // 对于每一个输入的整数
40 for (int i = 0; i < n; i++) {
41     int cmd = cmds[i];
42
43     // 检查输入的整数cmd是否在指定的范围内
44     if (cmd < -100 || cmd > 100) {
45         cout << "12345" << endl;
46         return 0;
47     }
48     // 如果输入的整数cmd等于l
49     if (cmd == lucky) {
50         // 如果cmd大于0, 那么位置p增加cmd+1
51         if (cmd > 0) {
52             p += cmd + 1;
53 }
```

```

54         } else if(cmd < 0){
55             // 否则, 位置p减少cmd-1
56             p += cmd - 1;
57         }
58     } else {
59         // 如果输入的整数cmd不等于L, 那么位置p增加cmd
60         p += cmd;
61     }
62     // 更新最大位置变量mp
63     mp = max(mp, p);
64 }
65
66 // 打印最大位置变量mp
67 cout << mp << endl;
68
69 return 0;
70 }

```

Java

```

1  import java.util.Scanner;
2
3  public class Main {
4      public static void main(String[] args) {
5          // 创建一个新的扫描器实例
6          Scanner sc = new Scanner(System.in);
7
8          // 读取用户输入的整数n和L
9          int n = Integer.parseInt(sc.nextLine());
10         int lucky = Integer.parseInt(sc.nextLine());
11
12         // 检查输入的整数n和L是否在指定的范围内
13         if (n < 1 || n > 100 || lucky < -100 || lucky > 100) {
14             System.out.println("12345");
15             return;
16         }
17
18         // 初始化位置变量p和最大位置变量mp
19         int p = 0;
20         int mp = 0;
21

```

```

41
42
43 // 读取用户输入的整数cmd
44 String[] cmds = sc.nextLine().split(" ");
45
46 // 对于每一个输入的整数
47 for (int i = 0; i < n; i++) {
48     int cmd = Integer.parseInt(cmds[i]);
49
50     // 检查输入的整数cmd是否在指定的范围内
51     if (cmd < -100 || cmd > 100) {
52         System.out.println("12345");
53         return;
54     }
55     // 如果输入的整数cmd等于L
56     if (cmd == lucky) {
57         // 如果cmd大于0, 那么位置p增加cmd+1
58         if (cmd > 0) {
59             p += cmd + 1;
60         } else if (cmd < 0) {
61             // 否则, 位置p减少cmd-1
62             p += cmd - 1;
63         }
64     } else {
65         // 如果输入的整数cmd不等于L, 那么位置p增加cmd
66         p += cmd;
67     }
68     // 更新最大位置变量mp
69     mp = Math.max(mp, p);
70 }
71
72 // 打印最大位置变量mp
73 System.out.println(mp);
74 }
75 }

```

javaScript

```

1  const readline = require('readline').createInterface({
2    input: process.stdin,
3    output: process.stdout
4  },

```

```
4  });
5
6  // 创建一个新的扫描器实例
7  let input = [];
8
9  readline.on('line', (line) => {
10     input.push(line);
11 });
12
13 readline.on('close', () => {
14     // 读取用户输入的整数n和l
15     let n = parseInt(input[0]);
16     let lucky = parseInt(input[1]);
17
18     // 检查输入的整数n和l是否在指定的范围内
19     if (n < 1 || n > 100 || lucky < -100 || lucky > 100) {
20         console.log("12345");
21         process.exit(0);
22     }
23
24     // 初始化位置变量p和最大位置变量mp
25     let p = 0;
26     let mp = 0;
27
28     // 读取用户输入的整数cmd
29     let cmds = input[2].split(" ").map(Number);
30
31     // 对于每一个输入的整数
32     for (let i = 0; i < n; i++) {
33         let cmd = cmds[i];
34
35         // 检查输入的整数cmd是否在指定的范围内
36         if (cmd < -100 || cmd > 100) {
37             console.log("12345");
38             process.exit(0);
39         }
40         // 如果输入的整数cmd等于l
41         if (cmd == lucky) {
42             // 如果cmd大于0, 那么位置p增加cmd+1
43             if (cmd > 0) {
44
```

```

45         p += cmd + 1;
46     } else if(cmd < 0){
47         // 否则, 位置p减少cmd-1
48         p += cmd - 1;
49     }
50 } else {
51     // 如果输入的整数cmd不等于l, 那么位置p增加cmd
52     p += cmd;
53 }
54 // 更新最大位置变量mp
55 mp = Math.max(mp, p);
56 }
57
58 // 打印最大位置变量mp
59 console.log(mp);
    });

```

Python

```

1  # 创建一个新的扫描器实例
2  import sys
3
4  # 读取用户输入的整数n和l
5  n = int(sys.stdin.readline())
6  lucky = int(sys.stdin.readline())
7
8  # 检查输入的整数n和l是否在指定的范围内
9  if n < 1 or n > 100 or lucky < -100 or lucky > 100:
10     print("12345")
11     sys.exit()
12
13 # 初始化位置变量p和最大位置变量mp
14 p = 0
15 mp = 0
16
17 # 读取用户输入的整数cmd
18 cmds = list(map(int, sys.stdin.readline().split()))
19
20 # 对于每一个输入的整数
21 for i in range(n):
22

```



```

22     cmd = cmds[i]
23
24     # 检查输入的整数cmd是否在指定的范围内
25     if cmd < -100 or cmd > 100:
26         print("12345")
27         sys.exit()
28     # 如果输入的整数cmd等于l
29     if cmd == lucky:
30         # 如果cmd大于0, 那么位置p增加cmd+1
31         if cmd > 0:
32             p += cmd + 1
33         elif cmd < 0:
34             # 否则, 位置p减少cmd-1
35             p += cmd - 1
36     else:
37         # 如果输入的整数cmd不等于l, 那么位置p增加cmd
38         p += cmd
39     # 更新最大位置变量mp
40     mp = max(mp, p)
41
42 # 打印最大位置变量mp
43 print(mp)

```

C语言

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  int main() {
6      // 创建一个新的扫描器实例
7      char line[1024];
8
9      // 读取用户输入的整数n和l
10     fgets(line, sizeof(line), stdin);
11     int n = atoi(line);
12     fgets(line, sizeof(line), stdin);
13     int lucky = atoi(line);
14
15     // 检查输入的整数n和l是否在指定的范围内
16

```

```

16 if (n < 1 || n > 100 || lucky < -100 || lucky > 100) {
17     printf("12345\n");
18     return 0;
19 }
20
21 // 初始化位置变量p和最大位置变量mp
22 int p = 0;
23 int mp = 0;
24
25 // 读取用户输入的整数cmd
26 fgets(line, sizeof(line), stdin);
27 int cmds[n];
28 char* token = strtok(line, " ");
29 for (int i = 0; i < n; i++) {
30     cmds[i] = atoi(token);
31     token = strtok(NULL, " ");
32 }
33
34 // 对于每一个输入的整数
35 for (int i = 0; i < n; i++) {
36     int cmd = cmds[i];
37
38     // 检查输入的整数cmd是否在指定的范围内
39     if (cmd < -100 || cmd > 100) {
40         printf("12345\n");
41         return 0;
42     }
43     // 如果输入的整数cmd等于L
44     if (cmd == lucky) {
45         // 如果cmd大于0, 那么位置p增加cmd+1
46         if (cmd > 0) {
47             p += cmd + 1;
48         } else {
49             // 否则, 位置p减少cmd-1
50             p += cmd - 1;
51         }
52     } else {
53         // 如果输入的整数cmd不等于L, 那么位置p增加cmd
54         p += cmd;
55     }
56 }

```

```
57 |         // 更新最大位置变量mp
58 |         if (p > mp) {
59 |             mp = p;
60 |         }
61 |     }
62 |
63 |     // 打印最大位置变量mp
64 |     printf("%d\n", mp);
65 |
66 |     return 0;
    }
```

完整用例

用例1

1,2,3,4,5,6,7,8,9,10

2

4

用例2

1,2,3,4,5

6

2

用例3

1,2,3,4,5

1

5

用例4

1,2,3,4,5,6,7,8,9,10

3

0

用例5

1
1
1

用例6

1,2,3,4,5,6,7,8,9,10
1
3

用例7

1,2,3,4,5,6,7,8,9,10
10
5

用例8

2,4,6,8,10,12
3
2

用例9

1,2
1
1

用例10

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
14
3

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

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