

Тренировки по алгоритмам 5.0 от Яндекса — Занятие 1 (Сложность, тестирование, особые случаи)

12 мар 2024, 20:13:55
старт: 1 мар 2024, 22:30:00
финиш: 12 мар 2024, 20:00:00
длительность: 10д. 21ч.
начало: 1 мар 2024, 22:30:00
конец: 12 мар 2024, 20:00:00

В. Футбольный комментатор

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Раунд плей-офф между двумя командами состоит из двух матчей. Каждая команда проводит по одному матчу «дома» и «в гостях». Выигрывает команда, забившая большее число мячей. Если же число забитых мячей совпадает, выигрывает команда, забившая больше мячей «в гостях». Если и это число мячей совпадает, матч переходит в дополнительный тайм или серию пенальти. Вам дан счёт первого матча, а также счёт текущей игры (которая ещё не завершилась). Помогите комментатору сообщить, сколько голов необходимо забить первой команде, чтобы победить, не переводя игру в дополнительное время.

Формат ввода

В первой строке записан счёт первого матча в формате $G_1:G_2$, где G_1 — число мячей, забитых первой командой, а G_2 — число мячей, забитых второй командой.
Во второй строке записан счёт второго (текущего) матча в аналогичном формате. Все числа в записи счёта не превышают 5.
В третьей строке записано число I , если первую игру первая команда провела «дома», или 2, если «в гостях».

Формат вывода

Выведите единственное целое число — необходимое количество мячей.

Пример 1

Ввод

0:0
0:0
1

Вывод

1

Пример 2

Ввод

0:2
0:3
1

Вывод

5

Пример 3

Ввод **Вывод**

0:2

6

0:3

2

Язык Kotlin 1.9.21 (JRE 21)

Набрать здесь

Отправить файл

```
1 fun main(args: Array<String>) {
2     val firstScore = readLine()!!.split(":").map { it.toInt() }
3     val secondScore = readLine()!!.split(":").map { it.toInt() }
4     val firstHomeGame = readLine()
5
6     val homeScore = if (firstHomeGame == "1") firstScore[0] else secondScore[0]
7     var guestScore = if (firstHomeGame == "1") secondScore[0] else firstScore[0]
8
9     var score = homeScore + guestScore
10    val lastScore = score
11
12    val enemyHomeScore = if (firstHomeGame == "1") secondScore[1] else firstScore[1]
13    val enemyGuestScore = if (firstHomeGame == "1") firstScore[1] else secondScore[1]
14
15    val enemyScore = enemyHomeScore + enemyGuestScore
16
17    while (true) {
18        if (score == enemyScore && guestScore > enemyGuestScore) {
19            println(score - lastScore)
20            return
21        }
22        if (firstHomeGame == "1") { // текущая -- в гостях
23            if ((guestScore > enemyGuestScore && score >= enemyScore) || score > enemyScore) {
24                println(score - lastScore)
25                return
26            }
27            guestScore++
28            score++
29        } else { // firstHomeGame == "2" // текущая -- дома
30            if (score > enemyScore) {
31                println(score - lastScore)
32                return
33            }
34            score++
35        }
36    }
37 }
```

Отправить

Предыдущая

Следующая