

Отбор на Школу Go разработки Ozon

7 июн 2020, 23:08:23
старт: 14 май 2020, 22:49:36
конец: 31 май 2020, 23:59:00

А. Уникальное число

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input-201.txt
Вывод	стандартный вывод или input-201.a.txt

На вход программе подается большое количество целых чисел. Все числа, кроме одного, имеют пару, причем может быть несколько одинаковых пар. Найдите число без пары.

Формат ввода

stdin десятичные числа по одному на каждой строке

Формат вывода

stdout десятичное число

Пример

Ввод	Вывод
1	3
2	
2	
1	
2	
3	
2	

Язык

Make

Набрать здесь

Отправить файл

1

Отправить

В. Теги

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В базе данных имеется таблица с товарами goods (id INTEGER, name TEXT), таблица с тегами tags (id INTEGER, name TEXT) и таблица связи товаров и тегов tags_goods (tag_id INTEGER, goods_id INTEGER, UNIQUE (tag_id, goods_id)).

Выведите id и названия всех товаров, которые имеют все возможные теги в этой базе.

БД - SQLite3. В качестве языка решения выберите make2.

Формат ввода

SQL-запрос.

Язык

Make

Набрать здесь

Отправить файл

1

Отправить

F. Сумма двух

Ограничение времени	1.5 секунд
Ограничение памяти	64Mb
Ввод	input.txt
Вывод	output.txt

Дано целое положительное число "target". Также дана последовательность из целых положительных чисел. Необходимо записать в выходной файл "1", если в последовательности есть два числа сумма, которых равна значению "target" или "0" если таких нет.

Формат ввода

5
1 7 3 4 7 9

Формат вывода

1

Примечания

Все числа используемы в задаче находятся в диапазоне $0 < N < 999999999$
Название входной файл: input.txt
Название выходной файл: output.txt

Язык

Make

Набрать здесь

Отправить файл

1

Отправить

D. Сложение чисел

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод
Вывод	стандартный вывод

Даны два неотрицательных числа А и В (числа могут содержать до 1000 цифр). Вам нужно вычислить их сумму.

Формат ввода

Первая строка ввода содержит числа А и В, разделенные пробелом

Формат вывода

Результат сложения двух чисел нужно вывести на отдельной строке.

Пример 1

Ввод	Вывод
1 2	3

Пример 2

Ввод	Вывод
199 1	200

Язык

Make

Набрать здесь

Отправить файл

1

Отправить

Е. 2 канала

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	см формат ввода
Вывод	см формат вывода

Необходимо написать функцию `func Merge2Channels(f func(int) int, in1 <-chan int, in2 <- chan int, out chan<- int, n int)` в `package main`.

Описание ее работы:

n раз сделать следующее

- прочитать по одному числу из каждого из двух каналов in1 и in2, назовем их x1 и x2.
- вычислить $f(x1) + f(x2)$
- записать полученное значение в out

Функция `Merge2Channels` должна быть неблокирующей, сразу возвращая управление.

Функция `f` может работать долгое время, ожидая чего-либо или производя вычисления.

Формат ввода

- Количество итераций передается через аргумент `n`.
- Целые числа подаются через аргументы-каналы `in1` и `in2`.
- Функция для обработки чисел перед сложением передается через аргумент `f`.

Формат вывода

Канал для вывода результатов передается через аргумент `out`.

Примечания

Отправлять задачу необходимо под компилятором `Make`. Решения, выдающие неверный ответ, могут по техническим причинам получать вердикт `Runtime Error`. Медленные решения получают вердикт `Idleness Limit`, стоит рассматривать это как превышение времени исполнения.

Язык

```
1 package main
2
3 func resolve(r chan<- int, x int, f func(int) int) {
4     r <- f(x)
5 }
6
7 func read(xStore chan<- int, in <-chan int, n int) {
8     for i := 0; i < n; i++ {
9         xStore <- <-in
10    }
11    close(xStore)
12 }
13
14
15 func Merge2Channels(f func(int) int, in1 <-chan int, in2 <-chan int, out chan<- int, n int) {
16     go func() {
17         RESULT := make(chan int, 100)
18         var xx1 []int
19         var xx2 []int
20         defer func() {
21             var result []int
22             for r := range RESULT {
23                 result = append(result, r)
24             }
25         }()
26         x1ch := make(chan int, n)
27         x2ch := make(chan int, n)
28         go read(x1ch, in1, n)
29         go read(x2ch, in2, n)
30
31         rx1ch := make(chan int, n)
32         rx2ch := make(chan int, n)
33         for i := 0; i < n; i++ {
34             x1 := <- x1ch
35             x2 := <- x2ch
36             xx1 = append(xx1, x1)
37             xx2 = append(xx2, x2)
38             go resolve(rx1ch, x1, f)
```