

# Стажировка весна-лето 2023 | бэкенд

7 фев 2023, 16:19:21

старт: 6 фев 2023, 11:48:41

финиш: 6 фев 2023, 16:48:41

длительность: 05:00:00

начало: 1 фев 2023, 00:00:00

## В. Горе от перфекционизма

	Все языки	GNU C++20 10.2	Clang14 C++20
Ограничение времени	3 секунды	1 секунда	1 секунда
Ограничение памяти	256Mb	256Mb	256Mb
Ввод	стандартный ввод или input.txt		
Вывод	стандартный вывод или output.txt		

**Объявление:** если у вас есть **жалобы / претензии / вопросы** насчет задач, то советуем для начала ознакомиться с [расширенной инструкцией](#), содержащей в том числе ответы на самые частые вопросы.

**Начало условия:** К Новому-преновому году работники Тындекса построили  $N$  ледяных скульптур,  $i$ -я скульптура состоит из  $a_i$  килограмм льда.

Но они не посоветовались с Кузей! А ведь Кузя знает, что **идеальная** скульптура состоит из ровно  $X$  килограмм льда, не больше и не меньше.

Новый-преновый год уже совсем скоро, до него осталось всего  $T$  минут. За одну минуту Кузя может выбрать одну скульптуру и добавить или удалить ровно 1 килограмм льда из неё.

Вас, как отличника художественной школы, Кузя просит найти максимальное количество идеальных скульптур в момент наступления праздника.

## Формат ввода

В первой строке вводятся три целых числа  $N, X, T$  ( $1 \leq N \leq 2 \cdot 10^5; 0 \leq X \leq 10^9; 0 \leq T \leq 3 \cdot 10^{14}$ ) — количество скульптур, идеальное количество льда в скульптуре и оставшееся количество минут до наступления праздника.

Во второй строке вводятся через пробел  $N$  целых чисел  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — количество килограмм льда в  $i$ -й скульптуре.

## Формат вывода

В первой строке выведите целое число  $K$  ( $0 \leq K \leq N$ ) — максимально возможное количество идеальных скульптур в момент наступления праздника.

Во второй строке выведите через пробел  $K$  различных целых чисел  $b_i$  ( $1 \leq b_i \leq N$ ) — номера скульптур, которые будут идеальными в момент наступления Нового-пренового года.

Скульптуры нумеруются с 1 в порядке ввода.

Если **оптимальных** ответов несколько, то выведите **любой** из оптимальных.

## Пример 1

Ввод 

3 5 2

5 10 6

Вывод 

2

1 3

## Пример 2

Ввод	Вывод
5 19 32	2
36 10 72 4 50	2 4

Пример 3

Ввод	Вывод
4 25 10	0
1 10 42 9	

Примечания

Пояснение к **первому** тестовому примеру.

До нового года остаётся 2 минуты, а идеальная скульптура должна содержать ровно 5 килограмм льда.

- 1. Первая скульптура идеальна сразу, поэтому Кузя не тратит времени на её исправление.
- 2. Кузя может сделать идеальной третью скульптуру за  $|6 - 5| = 1$  минуту. После этого у него в запасе останется  $2 - 1 = 1$  минута.
- 3. Кузя не сможет сделать идеальной вторую скульптуру, так как на её исправление необходимо  $|10 - 5| = 5$  минут.

Пояснение ко **второму** тестовому примеру.

До нового года остаётся 32 минуты, а идеальная скульптура должна содержать ровно 19 килограмм льда.

Рассмотрим, сколько требуется времени на «идеализацию» фигур:

- 1.  $|19 - 36| = 17$  минут;
- 2.  $|19 - 10| = 9$  минут;
- 3.  $|19 - 72| = 53$  минуты;
- 4.  $|19 - 4| = 15$  минут;
- 5.  $|19 - 50| = 31$  минута.

Итого получаются три возможных сценария с двумя идеальными фигурами:

- 1. Первая и вторая за  $17 + 9 = 26$  минут;
- 2. Первая и четвертая за  $17 + 15 = 32$  минуты - обратите внимание, что в данном сценарии Кузя потратит полностью время, оставшееся до события;
- 3. Вторая и четвертая за  $9 + 15 = 24$  минуты.

Хотя Кузя может сделать идеальной пятую фигуру, но на неё одну потребуется почти всё время (31 из 32 минут), поэтому Кузя не рассматривает такие сценарии.

Пояснение ко **третьему** тестовому примеру.

До нового года остаётся 10 минут, а идеальная скульптура должна содержать ровно 25 килограмм льда.

Кузя не успеет сделать ни одну из фигур идеальной, так как на каждую из них требуется больше, чем 10 минут:

- 1.  $|1 - 25| = 24 > 10$ ;
- 2.  $|10 - 25| = 15 > 10$ ;
- 3.  $|42 - 25| = 17 > 10$ ;
- 4.  $|9 - 25| = 14 > 10$ .

```

1 package main
2 import (
3     "fmt"
4 )
5
6 func main() {
7     var n, x, t, k int
8     var a, dif []int
9     var aa, min int
10    fmt.Scan(&n, &x, &t)
11    for i := 0; i < n; i++ {
12        fmt.Scan(&aa)
13        a = append(a, aa)
14        dif = append(dif, Abs(a[i] - x))
15        if (dif[i] == 0){
16            k++
17        }
18    }
19    for t > 0 {
20        min = 1000000000
21        for i := 0; i < n; i++ {
22            if (dif[i] < min && dif[i] != 0) {
23                min = dif[i]
24            }
25        }
26        for i := 0; i < n; i++ {
27            if (dif[i] == min) {
28                dif[i] -= min
29                t -= min
30                k++
31                if (t < 0){
32                    dif[i] += min
33                    k--
34                }
35            }
36        }
37    }
38 }

```

Отправить

Предыдущая

Следующая