

Яндекс. Тренировки по алгоритмам июнь 2021, занятие 7

G. Детский праздник

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

Организаторы детского праздника планируют надуть для него M воздушных шариков. С этой целью они пригласили N добровольных помощников, i -й среди которых надувает шарик за T_i минут, однако каждый раз после надувания Z_i шариков устает и отдыхает Y_i минут. Теперь организаторы праздника хотят узнать, через какое время будут надуты все шарiki при наиболее оптимальной работе помощников, и сколько шариков надует каждый из них. (Если помощник надул шарик, и должен отдохнуть, но больше шариков ему надувать не придется, то считается, что он закончил работу сразу после окончания надувания последнего шарика, а не после отдыха).

Формат ввода

В первой строке входных данных задаются числа M и N ($0 \leq M \leq 15000$, $1 \leq N \leq 1000$). Следующие N строк содержат по три целых числа - T_i , Z_i и Y_i соответственно ($1 \leq T_i$, $Y_i \leq 100$, $1 \leq Z_i \leq 1000$).

Формат вывода

Выведите в первой строке число T - время, за которое будут надуты все шарiki. Во второй строке выведите N чисел - количество шариков, надутых каждым из приглашенных помощников. Разделяйте числа пробелами. Если распределений шариков несколько, выведите любое из них.

Пример 1

Ввод

```
1 2
2 1 1
1 1 2
```

Вывод

```
1
0 1
```

Пример 2

Ввод

```
2 2
1 1 1
1 1 1
```

Вывод

```
1
1 1
```

Язык

```

1 def lbin_search(helpers, target, len_helpers):
2     lf, rg = 0, 200*15000
3     while lf < rg:
4         mid = (lf + rg) // 2
5         total = 0
6         for i in range(len_helpers):
7             cycles, rems = divmod(mid, helpers[i][2])
8             total += cycles * helpers[i][1] + min(rems//helpers[i][0], helpers[i][1])
9
10        if total < target:
11            lf = mid + 1
12        else:
13            rg = mid
14
15    cur_target = target
16    answer = [0] * len_helpers
17    for i in range(len_helpers):
18        cycles, rems = divmod(lf, helpers[i][2])
19        answer[i] = min(cur_target, cycles * helpers[i][1] + min(rems//helpers[i][0], helpers[i][1]))
20        cur_target -= answer[i]
21
22    return lf, answer
23
24 # считываем данные
25 M, N = map(int, input().split())
26 helpers = [None] * N
27
28 for i in range(N):
29     T, Z, Y = map(int, input().split())
30     # сохраняем время надувания шара, количество повторений до отдыха, длину цикла надувания и отдыха
31     helpers[i] = (T, Z, T * Z + Y)
32
33
34 count, count_list = lbin_search(helpers, M, N)
35 print(count)
36

```

Отправить

Предыдущая

Следующая