



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2283 - Diseño y Análisis de Algoritmos - 2^{do} semestre 2022

TAREA 0

Publicación: Lunes 22 de agosto.
GitHub Classroom: <https://classroom.github.com/a/9QpjXOog>
Entrega: **Domingo 28 de agosto 23:59 horas.**

Indicaciones

- La tarea **opcional** y estrictamente individual.
- La solución debe ser entregada en el archivo `t0.py` del repositorio privado asignado mediante GitHub Classroom para esta tarea. Se revisará el último *commit* subido antes de la entrega al repositorio. Se usará Python 3.10.X para la revisión.
- El *input* para el programa debe ser obtenido desde *standard input*. El *output* debe ser entregado mediante *standard output*.
- La corrección se realizará mediante *tests* automatizados acordes al formato de *input* y *output* especificado. Cada *test* tendrá un *timeout* según lo que se especifica como tiempo esperado.
- Un *test* se considerará **reprobado** en caso de que 1) dado el *input* el *output* sea incorrecto, 2) exista un error de *runtime* durante su ejecución, o 3) el *timeout* se cumpla durante su ejecución. En otro caso, el *test* se considerará **aprobado**.
- No se permite el uso de librerías externas a la librería estándar de Python *a priori*. Consultar en las [issues del repositorio oficial del curso](#) en caso de requerir una.
- **Para esta tarea especial no hay posibilidad de entregar con atraso.**
- En caso de aprobar todos los *tests* de corrección, el alumno obtendrá automáticamente una bonificación de **3 décimas** adicionales al promedio de tareas **PT**.

Problema

Emilia es una joven enérgica con muchos intereses, lo que hace que cada día sea un océano de posibilidades para ella, lleno de actividades emocionantes en las que está más que dispuesta a participar. Por desgracia, muchas de las actividades en las que participa Emilia se desarrollan en espacios cerrados, y por ello sus niveles de vitamina D están ligeramente por debajo de los ideales. Para ayudarla, un médico le recetó un suplemento vitamínico que debe tomar diariamente cada X minutos.

Emilia ha creado una aplicación para llevar el control de sus actividades. La principal característica de la aplicación es la programación de actividades. Cada actividad consta de un título y de las horas de inicio y finalización. La aplicación también permite crear recordatorios puntuales, que consisten simplemente en un título y el momento en que debe recibir una única notificación, y recordatorios recurrentes, que consisten en

un título, la hora de la primera notificación y la frecuencia con que debe repetirse el recordatorio. Después de comprar el suplemento, Emilia quiere añadir un recordatorio recurrente a la aplicación, que se repita cada X minutos, para asegurarse de que recibirá una notificación cada vez que sea el momento de tomar el suplemento. Lo ideal es que no tenga que tomar el suplemento mientras participa en una actividad. Con una agenda tan apretada, le resulta difícil encontrar el momento ideal para empezar a tomarlo.

Tu tarea es ayudar a Emilia a elegir el momento T ideal después de comprar el suplemento para tomarlo por primera vez. Se dice que un momento T es **ideal** si:

1. Es como máximo 8 horas después del momento en que Emilia compró el suplemento.
2. El número de veces que necesita tomar el suplemento que entra en conflicto con sus actividades es mínimo.
3. No hay ningún otro momento anterior a T que dé lugar al mismo número de conflictos.

Para mayor claridad, supongamos que Emilia necesita tomar el suplemento cada 30 minutos, lo compró un determinado día a las 10:00, y tiene una única actividad programada de 18:00 a 18:30 durante ese día. Si elige tomar el suplemento por primera vez inmediatamente, lo tomaría a las 10:00, 10:30, ..., 17:30, 18:00, 18:30, Así, tendría dos conflictos con su actividad (uno a las 18:00 y otro a las 18:30).

Aunque Emilia no puede esperar a que termine su actividad para tomar el suplemento por primera vez (debido al límite de 8 horas), puede reducir el número de conflictos esperando 7 horas y 59 minutos. De este modo, tomaría el suplemento a las 17:59, 18:29, 18:59, ..., con un único conflicto a las 18:29. Mucho mejor, ¿no? Sin embargo, esto no es ideal, porque hay una hora anterior que también produce un único conflicto, que es esperar 1 minuto y tomar el suplemento por primera vez a las 10:01, con un único conflicto a las 18:01. Como no es posible evitar completamente los conflictos, la hora ideal, en este caso, es tomar el suplemento por primera vez 1 minuto después de comprarlo.

Se te dará una lista de actividades, exportada desde la aplicación que escribió Emilia, que contiene información sobre todas sus actividades a partir del momento en que compró el suplemento. Emilia ya ha hecho un preprocesamiento de los datos y cada actividad se describe con el número de minutos desde que compró el suplemento y su duración en minutos.

Input

La primera línea contiene dos enteros N ($1 \leq N \leq 10^4$) y X ($1 \leq X \leq 720$), indicando que Emilia tiene N actividades próximas según su app, y que debe tomar su suplemento cada X minutos. Cada una de las siguientes N líneas describe una actividad con dos enteros S y D ($1 \leq S \leq 10^5$, $1 \leq D \leq 10^5$), representando que la actividad comienza S minutos luego de comprar el suplemento, y que su duración es de D minutos. Las actividades no se solapan, es decir, dado cualquier par de actividades, una de ellas termina estrictamente después que la otra, o vice versa.

Output

El output es una sola línea con dos enteros T y C , indicando respectivamente el momento ideal para tomar el suplemento por primera vez, expresado en minutos desde la compra del suplemento, y el número de conflictos que provocará la toma del suplemento en ese momento.

Tiempo esperado

Se espera que la solución se ejecute en un tiempo **menor o igual a 0.3 segundos** para cualquier instancia de input según las restricciones dadas.

Complejidad esperada

No hay restricciones de complejidad para este problema.

Ejemplos

Los siguientes ejemplos están ya cargados a GitHub Classroom con corrección automática mediante GitHub Actions. **Los tests para la corrección serán distintos a estos.**

Input
1 30 480 30
Output
1 1

Input
5 30 195 30 120 45 240 30 30 60 300 180
Output
451 1

Input
4 720 60 30 150 75 750 60 1500 60
Output
0 0

Input
2 720 1 479 482 298
Output
0 1