Concurso de Programación

Concurso I 2024 - Estructura de Datos

https://urjc-cp.github.io/urjc-cp/

Cuadernillo de problemas

URJC Escuela Técnica Superior de Ingeniería Informática

Realizado en la **Escuela Técnica Superior de Ingeniería Informática (URJC)**
16 de febrero de 2024

Índice

Autores de los problemas:

- Isaac Lozano Osorio (Universidad Rey Juan Carlos)
- Sergio Salazar Cardenas (Universidad Rey Juan Carlos)

Tiempo: 1,8 segundos

A
Vuelta a la rutina

Juan va a comenzar su segundo año de universidad, y con él llega una nueva oleada de sufrimiento: que si nuevas asignaturas y profesores, que si trabajos, que si papeleo, un montón de cosas pendientes por hacer, ... pfff. Solo de pensarlo se pone malo.

Para conseguir que la vuelta a la rutina sea lo menos dolorosa posible, se le ha ocurrido hacer una aplicación que le ayude a llevar la cuenta de todas las tareas que tiene pendientes. De esta forma, nunca se dejará nada olvidado, y le ayudará a evitar repetir la misma tarea múltiples veces, empezando así el curso con buen pie. ¿Podrías ayudarle?

Entrada

En la primera línea encontrarás dos números, N y M, separados por un espacio, que indican, respectivamente, el número de tareas diferentes que Juan tiene que hacer y el número de tareas que Juan ha realizado ya.

En cada una de las siguientes N líneas, encontrarás el nombre de cada tarea que Juan debe realizar. El nombre de la tarea cumple "[a-zA-Z]1,32"(solo caracteres a-z, A-Z y espacio, entre 1 y 32 caracteres). Se garantiza que todas las tareas N_i a realizar son únicas.

En cada una de las siguientes M líneas, encontrarás el nombre de cada tarea que Juan ha realizado. Como Juan es un poco despistado, es posible que haga la misma tarea varias veces. Se garantiza que todas las tareas que Juan ha realizado aparecen en la lista de tareas a realizar.

Salida

Para cada caso de prueba, deberás decir "PONTE A TRABAJAR", si tras consumir toda la entrada tiene tareas pendientes; "PUEDES DESCANSAR", si todas las tareas han sido realizadas una sola vez; o "TRABAJAS DEMASIADO", si todas las tareas han sido realizadas al menos una vez, pero alguna tarea ha sido hecha múltiples veces.

Entrada de ejemplo

```
1 2
Estudiar
Estudiar
Estudiar
```

Salida de ejemplo

```
TRABAJAS DEMASIADO
```

Entrada de ejemplo

```
4 4
Estudiar
Lavar los platos
Recoger habitacion
Hacer la compra
Hacer la compra
Lavar los platos
Hacer la compra
Recoger habitacion
```

Salida de ejemplo

```
PONTE A TRABAJAR
```

Límites

- $1 \leq N, M \leq 1000000$
- $1 \leq$ Letras de tarea ≤ 32

Tiempo: 1 segundo

B
Victoria Magistral

Todos sabemos que en Fortnite te pueden derribar en cuestión de milésimas. Tu amigo, Robius, necesita saber cuántos disparos tiene que dar a su rival para derrotarlo, es decir, sabiendo la vida que tiene el rival y el daño que hace un disparo, cuánto es necesario para vencerle. Pero como todos sabemos, existen diferentes armas y algunos rivales pueden tener más o menos vida que otros, por lo que para él es un lio y nos ha pedido que le ayudemos realizando un programa.

Entrada

Un número N con el número de casos, N casos que constituyen dos números V (vida que tiene el rival) y D (daño que hace la bala).

Salida

Número de disparos que tiene que realizar Robius para vencer a su rival. Todos los números son divisibles.

Entrada de ejemplo

```
2
200 10
200 100
```

Salida de ejemplo

```
20
2
```

Límites

- $1 \leq N, V, D \leq 10000$

Tiempo: 1 segundo

C
Tramitando Papeleo

La tramitación de papeles es ciertamente una lotería. Unas veces, entregas un documento y te dan respuesta en el mismo día. Otras, pueden pasar semanas ¡Incluso meses!

Una mañana decides pasarte por la oficina de tramitación, y te has dado cuenta en cuál es su funcionamiento: Cada vez que un nuevo papel entra en la oficina, se deja al final de todo, el último, y van cogiendo papeles de la mesa para tramitarlos, poniendo los nuevos debajo. Sin embargo, el orden de entrada no corresponde siempre con el de tramitación, parece que hay algo más...

Finalmente, un día pasando de refilón te das cuenta al oir "No, ¡este papel me va a quitar mucho tiempo, lo dejo para luego!" ¡Y ves que lo pone de nuevo al final del todo! Eso cuadra más. Y no solo eso, parece que también hacen lo mismo con algunos papeles de forma aleatoria.

Con toda la información, quieres hacer un programa que calculará, cuanto tardará la oficina de tramitación según los papeles que entren.

Según has observado, sabes que para cualquier determinado tiempo T, si se encuentran con un papel que tarda más de $T/5$ en procesarse, lo dejarán para luego (Excepto cuando $T \leq 10$, que están empezando y tienen más ganas de trabajar), y que también harán lo mismo con algunos papeles específicos, tarden lo que tarden, pero solo si no lo han visto antes (en estos casos da igual que $T \leq 10$, lo pospondrán de todas formas). El proceso de coger un papel es instantáneo, pero el de dejarlo para luego tardará un tiempo fijo de $T = 10$.

Entrada

La entrada corresponderá a un único caso de prueba, comenzando por una línea con un número N de papeles, ordenados según tiempo de entrada a la oficina de tramitación. Corresponderá, pues una línea con N papeles, donde cada papel vendrá identificado por un número S, la cantidad de tiempo que se tarda en tramitar. Para mayor facilidad, ningún papel tardará lo mismo en tramitarse que otro. Finalmente, acompañará una línea con un número P de papeles que, tarden lo que tarden, dejarán para después. Seguirán P líneas con el tiempo S que se tarda en tramitar cada papel de este tipo.

Salida

Se debe imprimir un único número con el tiempo que se tarda en tramitar todos los papeles, siguiendo el sistema de la oficina de tramitación.

Entrada de ejemplo

```
6
3 2 4 1 6 10
1
2
```

Salida de ejemplo

```
66
```

Ejemplo

- La oficina comienza con T=0.
- Se encuentran un papel de 3, como estamos en T $\leq$ 10, se tramita. T = 3.
- Se encuentran un papel de 2, como está incluido en la lista de papeles para después, se continúa con el siguiente y no se tramita (dejándolo para más tarde). T = 13.
- Se encuentra un papel de 4, que tarda más en tramitarse que 13/5, así que se deja para después. T = 23.
- Se encuentran un papel de 1, que se tramita. T = 24.
- Papel de 6, que tarda mas en tramitarse que 24/5, se deja para después. Sucede lo mismo con 10. T = 44.
- A continuación se encuentran de nuevo los papeles de 2, 4, 6 y 10 (En ese orden), que son todos tramitados, ya que 2 se ha dejado para después previamente y todos cumplen la condición de tramitarse en menos de T/5.
- T final = 66.

Límites

- $1 \leq N \leq 10.000$
- $1 \leq S \leq 100.000$
- $1 \leq P \leq 100$

Tiempo: 1 segundo

D
Envíos Prioritarios

Amazon está pensando añadir un nuevo tipo de subscripción a su servicio de compras online. El nuevo servicio lleva las siglas de "P2W", nadie está seguro del significado que llevan detrás.
Los usuarios que opten por pagar los 100 euros que cuesta podrán hacer que sus envíos se salten posiciones en la cola de salida de los centros de reparto; en concreto, saldrán antes que cualquier producto que no se haya pedido con este servicio.

Entrada

La entrada contendrá un único caso de prueba.
El caso empezará con un número, N, que indicará el número de eventos que sucederán a lo largo de una semana en el centro de reparto. Cada evento ocupará una línea y puede tener uno de los siguientes formatos:

- "P2W" o "NOR" - indicando la llegada de un paquete "P2W" o normal a la cola de reparto.
- "R C" - indicando que va a salir un camión con capacidad para C paquetes del centro de reparto.

Tanto en la cola de reparto como en el camión los paquetes se ordenan por ser "P2W" o no y luego por su orden de aparición.

Salida

Por cada camión que sale tienes que imprimir los identificadores de los paquetes que se va a llevar el camión (sin rebasar su capacidad), separados por un espacio. El camión puede salir sin llenarse e incluso vacío si no hay paquetes suficientes esperando en la cola. Los paquetes se identifican por su orden de aparición en la entrada, empezando en el 0.

Entrada de ejemplo

```
8
NOR
NOR
P2W
NOR
R 2
P2W
R 5
NOR
```

Salida de ejemplo

```
2 0
4 1 3
```

Límites

- $1 \leq N \leq 200000$
- $1 \leq C \leq 100$

Tiempo: 1 segundo

E
Cifrado del César

Para poder enviar mensajes cifrados, el César inventó el conocido como "Cifrado del César". En este cifrado, primero enviaba una paloma mensajera con la clave de cifrado y después enviaba una paloma por cada palabra que querían cifrar. Por limitaciones de la época, solo podían enviar palabras con letras minúsculas y sin caracteres especiales. El cifrado se hace palabra a palabra, sumando a cada carácter la clave módulo 26 (el número de letras que hay en el alfabeto). Por ejemplo, la palabra "holaz" con clave 2 se cifra como "jqncb". Implementa un programa que, dada una clave, una palabra y otra cifrada, compruebe si la palabra se corresponde o no con su cifrado.

Entrada

La primera línea contendrá un entero N que indica el número de palabras del mensaje. La segunda línea contendrá un entero K que indica la clave de cifrado. A continuación, las siguientes N líneas contendrán dos cadenas de caracteres W y C, que se corresponden con la palabra original y la cifrada, respectivamente.

Salida

Por cada pareja de palabras W y C, se deberá imprimir "SI" si C es el cifrado de W con la clave K o "NO" en caso contrario.

Entrada de ejemplo

```
1
2
holaz jqncb
```

Salida de ejemplo

```
SI
```

Límites

- $0 \leq N \leq 10000$
- $0 \leq M \leq 100000$

Tiempo: 1 segundo

F
El juego de la oca

El juego de la oca es un juego de mesa tradicional que permite jugar en tiempo real a través de una interfaz táctil con varios jugadores. En este juego cada jugador mueve su ficha siguiente un recorrido en espiral hasta que atraviese todas las casillas. Queremos saber si serías capaz o si te atreves a generar ese patrón en espiral para tableros cuadrados de diferentes tamaños. La primera casilla arriba a la izquierda, empieza con el número 1 y a partir de ella se numeran el resto de casillas siguiendo la dirección de las agujas del reloj.

Entrada

La entrada contiene un único número N indicando que el tamaño del tablero es N^2.

Salida

Se debe imprimir el tablero con sus casillas numeradas en la forma de N líneas, cada una con N números separados por un único espacio.

Entrada de ejemplo

```
4
```

Salida de ejemplo

```
1 2 3 4
12 13 14 5
11 16 15 6
10 9 8 7
```

Límites

- $1 \leq N \leq 15$

Tiempo: 1 segundo

G
Teclado Bancario

No es misterio para nadie que algunas personas se tardan más en teclear algunas palabras que otras, es el problema que nos trae hoy aquí.

Luis, un director de recursos humanos de una importante entidad bancaria, piensa en el bienestar de sus clientes, sabe que a ellos les frustra mucho cuando alguien está detrás de un ordenador escribiendo una a una las letras, mirando a cada rato el teclado para saber donde está la siguiente, en promedio, se tardan 100ms multiplicado por la cantidad de teclas que debe recorrer el dedo en busca de la siguiente. Desafortunadamente, Luis no ha tenido suerte y se aproximan días muy duros para el banco, necesita contratar personas ya, por lo que está intentando pensar en distintas distribuciones de teclados para que la gente escriba lo más rápido posible.

Probando con el qwerty, azerty e incluso el dvorak no parece solucionar mucho, por lo que ya cansado, ha llamado a una empresa para fabricar el teclado óptimo para su banco, sin embargo, debe garantizar que lo será, por lo que ha llevado esta pregunta hasta ti. Luis te proporcionará la disposición de un teclado, distribuido en una matriz de 4 filas por 9 columnas, con los números del 0 al 9 y con las letras de la a a la z del alfabeto inglés (36 teclas en total), por simplicidad, se asume que los trabajadores no usarán nunca acentos o caracteres especiales.

En un teclado qwerty, por ejemplo, si hemos de escribir retrete, un empleado promedio tardaría 100(re)+200(et)+100(tr)+100(re)+200(et)+200(te) ms para un total de 900ms!

Entrada

Las primeras 4 líneas contendrán 9 caracteres denotando la distribución del teclado. Después, un número N denotando las palabras que escribirá el empleado, después, N líneas con una palabra S_i

Salida

Para cada caso debes imprimir cuanto se tarda en escribir un empleado cada palabra S_i, suponiendo que es un empleado promedio.

Entrada de ejemplo

```
123456789
qwertyuio
asdfghjkl
zxcvbnmp0
3
retrete
adabyron
clasificatorio
```

Salida de ejemplo

```
900
2400
6200
```

Límites

- $1 \leq N \leq 1000$
- $1 \leq S \leq 100$

Tiempo: 1 segundo

H
Problema del cambio de monedas

La historia de las máquinas expendedoras se remonta al siglo I de nuestra era en Egipto, la primera de ellas consistía en un artilugio mecánico que al insertarle una moneda dispensaba agua bendita en los templos. Ahora tenemos máquinas más complicadas capaces de aceptar de cantidades arbitrarias de dinero, cobrar, y devolver el cambio restante.

En este problema se pide desarrollar un algoritmo que a partir de un importe determine cómo devolver su cambio. Se tendrá en cuenta que se debe devolver dicho importe utilizando el mínimo número de monedas posibles. El sistema monetario que utilizaremos es $M = 500, 200, 100, 50, 20, 10, 5, 2, 1$, en dicho orden. Además, una máquina real tiene un número limitado de monedas de cada tipo.

Entrada

En la primera línea se encuentra un número T que nos indica el número de casos a procesar. En la segunda línea nos encontramos 9 números $X = X_1, X_2, ..., X_9$ en el que cada valor X_i se corresponde con el número de unidades disponibles de la moneda M_i. A continuación, nos encontramos T líneas, cada una un número C con la cantidad de dinero a devolver.

Salida

La salida contiene T líneas, una por cada caso, que contienen número mínimo U de monedas que devolvería la máquina, partiendo siempre para cada caso desde el número inicial de monedas. Si es imposible devolver dicha cantidad se escribirá -1.

Entrada de ejemplo

```
10
1 10 1 1 0 1 0 0 30
0
10
9
31
40
41
223
723
2690
2691
```

Salida de ejemplo

```
0
1
9
22
31
-1
15
16
44
-1
```

Límites

- $1 \leq T \leq 10^6$
- $0 \leq X_i \leq 10^9$
- $0 \leq C \leq 10^9$
- $-1 \leq U \leq 10^9$