

Nivel 2do Año
Problem Set

11 de Noviembre de 2022

Problema 1: Parientes

SPOJ code: TMPARI

Un día lluvioso, Pedro, estaba aburrido en su habitación y jugando con números se le ocurrió agrupar a los números por parentesco para poder pasar el rato y no aburrirse, así que escribió muchos números en pedazos de papel y los mezcló, luego pensó un número con todos sus dígitos iguales y comenzó a sacar papelitos de la bolsa.

Definió los siguientes parentescos para los números y les dio el siguiente significado:

- Nietos: son todos los números que tienen 2 o menos dígitos iguales a los suyos.
- Hijos: son los que tienen 1 dígito igual al suyo menos que él.
- Hermanos: serán los números que tengan la misma cantidad de dígitos iguales a los suyos.
- Padres: serán los números que tengan un dígito igual al suyo más que él.
- Abuelos: serán los números que tengan dos dígitos igual al suyo más que él.
- Bisabuelos: serán los números que tengan tres o más dígitos iguales a los suyos más que él.

Entonces agrupó los números y obtuvo la siguiente lista con el número 33:

- Nietos: 1 (200)
- Hijos: 2 (243, 300)
- Hermanos: 2 (4533, 1353)
- Padres: 2 (3233, 13433)
- Abuelos: 1 (33433)
- Bisabuelos: 1 (3433323)

Ahora Pedro quiere que le ayudes a escribir un algoritmo que, dado un número y una lista de números, realice la agrupación por él.

Entrada

El primer valor es un número entero con una **cantidad** de dígitos entre 1 y 17, los cuales serán todos iguales. En la línea siguiente habrá una lista de N valores con $1 \leq N \leq 2^8$ y los valores de la lista van entre 1 y 2^{64}

Salida

La salida consta de la lista de parientes del número dado, indicando la cantidad correspondiente y los números entre paréntesis separados por espacios, si hay cero parientes solo se mostrará el 0 sin paréntesis. Si el primer número ingresado contiene solo un dígito se mostrará un mensaje que dice ERROR.

Ejemplo

Entrada 1

```
22
2243 3020 45323 32323 132422 13523 323222 2002
```

Salida 1

```
Nietos: 0
Hijos: 3 (3020 45323 13523)
Hermanos: 3 (2243 32323 2002)
Padres: 1 (132422)
Abuelos: 1 (323222)
Bisabuelos: 0
```

Entrada 2

```
000
2030 3000 05000 30300 130202 13530 300000 2220 000000
```

Salida 2

```
Nietos: 2 (2220 13530)
Hijos: 2 (2030 130202)
Hermanos: 2 (3000 30300)
Padres: 1 (05000)
Abuelos: 1 (300000)
Bisabuelos: 1 (000000)
```

Entrada 3

```
0
203 300 0500 3030 13022 1353 30000 222 00000
```

Salida 3

```
ERROR
```

Credit for this problem:**Matias Wettstein**

Problema 2: Caminata con Exclamaciones

SPOJ code: CAMINEX

Andrés y sus amigos son fanáticos del senderismo, es decir de realizar largas caminatas de un día o dos en ambientes naturales y disfrutando del paisaje. Según como se levantó el día de la caminata, Andrés puede cansarse con mayor o menor facilidad al subir las pendientes. También, como es muy ansioso, tiende a aburrirse cuando el camino es llano, es decir cuando por cierto trecho el camino no tiene subidas ni bajadas. Finalmente, disfruta mucho cuando el camino desciende, porque su andar toma más velocidad con la pendiente.

Sus amigos conocen bien sus exclamaciones cuando salen de excursión. Cuando el camino sube continuamente, llega un momento en el cual dice: "Me canso!". Ese momento depende de la energía que tenga ese día. Por otra parte, cuando el camino es recto exclama "Me aburro!", dependiendo de cuánta ansiedad tenga ese día. También, exclama "Weeee!" cuando está bajando una pendiente y ya tomó cierta velocidad, pero demora más en decirlo según cuán triste se siente.

Conociendo la energía, ansiedad y tristeza con las cuales Andrés encara la caminata, y las características de la senda a recorrer, ellos le piden que desarrolle un programa que les permita predecir las exclamaciones que saldrán de los labios de Andrés durante el camino.

Entrada

Incluye entre 0 y 100 excursiones, cada una representada por 2 líneas de la entrada:

La primera línea contiene 3 números enteros entre 1 y 10, separados por un espacio entre sí, que indican la *energía*, *ansiedad* y *tristeza* de Andrés al iniciar la caminata.

La segunda línea de cada excursión incluye la descripción de las características del trayecto, representada con un string *S* (cuya longitud varía entre 1 y 10000) con los siguientes caracteres:

- `'/'` ascenso (barra),
- `'-'` camino recto (guión) y
- `'\'` descenso (contrabarra).

Las distintas excursiones están separadas entre sí por una línea en blanco.

El listado de excursiones termina cuando los números correspondientes a la *energía*, *ansiedad* y *tristeza* de Andrés son todos 0 (cero).

Salida

Para cada excursión debe indicar en la primera línea el número de excursión y en las líneas siguientes las exclamaciones de Andrés durante la caminata (ver ejemplos).

Las exclamaciones se producen cada vez que se repiten en forma consecutiva los ascensos, caminos rectos o descensos igualando a su límite de energía, ansiedad o tristeza, respectivamente. Luego de cada exclamación Andrés se renueva (vuelve a su estado inicial del comienzo de la excursión).

Coloque una línea en blanco entre excursiones consecutivas.

Ejemplo

Entrada

```
2 3 4
///--\/----\\\--/////-----\\\\

5 1 3
/\/\\\--//////-\\

0 0 0
```

Salida

```
Excursion #1
Me canso!
Me aburro!
Me canso!
Me canso!
Me aburro!
Weeee!

Excursion #2
Weeee!
Me aburro!
Me aburro!
Me canso!
Me aburro!
```

Credit for this problem: **Pablo Marchetti**

Problema 3: Doble Agente

SPOJ code: TMDBLA

Hace unos meses se descubrió que un agente de inteligencia estaba compartiendo con un país rival coordenadas de edificios y lugares secretos del país, para transmitir estos mensajes se vio que utilizaba un sistema de cifrado bastante rudimentario pero efectivo para ojos inhábiles.

La Agencia le solicita a su equipo que cree un algoritmo que descifre los códigos que se interceptaron para descubrir qué zonas están en peligro.

La forma de encriptar los mensajes es la siguiente:

Es un texto compuesto solamente por letras mayúsculas de "A" a "Z", sin considerar otras letras (por ej. "Ñ"), aparentemente al azar, pero cuando en la cadena aparece una A, la siguiente letra representa el valor que ocupa en el abecedario (AA = 1, AB = 2, etc.).

Además, se descubrieron algunas excepciones:

- Si ocurre dos veces el mismo patrón separado por una Q esto representará 0, por ejemplo, ABQAB, como AB se repite es 0.
- En caso de que se encuentren dos P seguidas esto indicará la presencia de un punto (.).
- Si se encuentran dos E seguidas, esto representará un espacio ( ). Los espacios de más no se tendrán en cuenta.
- Por último, en caso de existir dos G seguidas estas representarán un guión (-).

Cada guión precede un dígito y cada punto precede y antecede un dígito, ambos no necesariamente adyacentes, es decir, pueden aparecer letras intermedias sin significado.

RECUERDA: las coordenadas van de a pares, en caso de que el mensaje cuente con un número impar de coordenadas o ninguna la entrada se invalidará.

Entrada

Consiste en una única línea que contiene un String de letras en el que puede haber 0 a N coordenadas. La entrada no contiene símbolos, números o espacios en blanco. El mensaje no contará con más de 1.000.000 de caracteres.

Salida

La salida deberá indicar el o los pares de coordenadas entre paréntesis y separados por espacios. En caso de ser una entrada que tenga un número impar de coordenadas o ninguna deberá devolver FALSO.

Ejemplos

Entrada 1

```
VBGGACAANBVFPPAFABACABJKIUAHAHUKEECSXGGAFBCAJQAJPPAFAHAIAEAGADQAD
```

Salida 1

```
(-31.623288 -60.689570)
```

Entrada 2

```
VBGGACAANBVFPPAFABACABJKIUAHAHUK
```

Salida 2

```
FALSO
```

Entrada 3

```
VBGGACAANBVFPPAFABACABJKIUAHAHUKEECSXGGAFBCAJQAJPPAFAHAIAEAGADQADEEVBAFA
ANBVFPPAFABACABJKIUUKEECSXACBCADPPAFAIAIAEAG
```

Salida 3

```
(-31.623288 -60.689570) (61.6232 34.69957)
```

Entrada 4

```
VBGGACAANBVFPPAFABACABJKIUAHAHUKEECSXGGAFBCAJQAJPPAFAHAIAEAGADQADEEVBAFA
ANBVFPPAFABACABJKIUUKEE
```

Salida 4

```
FALSO
```

Credit for this problem:**Matias Wettstein**

Problema 4: Tesoro Maya

SPOJ code: TESORO

Henry es un arqueólogo, profesor y aventurero en busca del tesoro de Pascal I, un antiguo gobernante del Imperio Maya.

Luego de años de investigación, Henry descubrió inscripciones que indicarían la ubicación del tesoro. Sin embargo, se encontró con dos problemas. El primero de ellos es que las inscripciones contienen números mayas (es decir, según el sistema numérico maya). El segundo problema es que Pascal I, en su afán de engañar a los saqueadores, incluyó secuencias de símbolos que aparentan ser números pero no lo son.

El último hallazgo de Henry fue que la suma de los números mayas obtenidos lo guiaría a la posición exacta de las riquezas que desde hace siglos permanecen ocultas.

La civilización maya utilizó un sistema numérico posicional vigesimal (base 20) que constaba de los siguientes símbolos: punto, raya y un caracol para el cero. En la Tabla 1 se muestran los números mayas y decimales de 0 a 19.

	0		1		2		3		4
	5		6		7		8		9
	10		11		12		13		14
	15		16		17		18		19

Tabla 1. Números mayas y decimales de 0 a 19.

Para números mayores a 19 se emplean niveles. El valor decimal de un número maya nivel N se obtendrá empleando: 20^{n-1}, 20^{n-2}, … 20^{0} (ver Tabla 2).

x 8000								
x 400								
x 20								
x 1								
	20	21	41	47	161	400	401	8000

Tabla 2. Ejemplo de números mayas y decimales mayores a 19.

Para este problema un número maya será representado con los caracteres punto (“.”), guión medio (“-”) y cero (“0”), con cada nivel separado por un espacio (“ ”). Cualquier secuencia de caracteres que no corresponda a un número maya será considerada no válida.

Entrada

La entrada comienza con una línea que contiene un único número S que indica la cantidad de casos de prueba, con 1 ≤ S ≤ 50 y luego una línea en blanco. A continuación, para cada caso de prueba la primera línea posee un único número N que corresponde a la cantidad de números mayas, con 1 ≤ N ≤ 1000. Cada una de las siguientes N líneas contienen un número maya M o una entrada no válida. Con 0 ≤ M ≤ 1.000.000.000 (valor decimal). Luego de cada caso de prueba hay una línea en blanco.

Salida

Por cada caso de prueba la salida tiene dos líneas: la primera posee un único número que corresponde a la sumatoria (de ese caso) y la segunda línea contiene un único número que indica la cantidad de entradas no válidas.

Ejemplo

Entrada

```
3

3
...
-
....---

5
. 0 -.
. 0 0
0 .
- .-
0 0 0

4
....--- ..--- ....--
..-.. 0
.
. 0 .
```

Salida

```
27
0
506
3
8356
1
```

Credit for this problem: **Cristian Martinez**

Problema 5: Números Afines

SPOJ code: NAFINES

La función σ(N) indica la sumatoria de los divisores de N (excepto el propio N). Por ejemplo, 40 tiene los divisores 1, 2, 4, 5, 8, 10 y 20, siendo σ(40) = 50.

La función σ(N) está relacionada con la amistad entre números. Pero como ocurre con las personas, los números pueden tener *afinidad* entre sí, sin llegar a un vínculo de amistad. El grado *afinidad* es un número entero entre 1 (máxima afinidad) y 10 (mínima afinidad). Un par de números N y M son *números afines* si cumplen con:

- $M = \sigma(N)$
- $1 \leq |N - \sigma(M)| \leq 10$

Los números 40 y 50 son *números afines*. Considerando N = 40,

- $M = \sigma(N) = \sigma(40) = 50$
- $1 \leq |N - \sigma(M)| \leq 10 \Rightarrow \quad 1 \leq |40 - \sigma(50)| \leq 10 \quad \Rightarrow$
  $1 \leq |40 - 43| \leq 10 \Rightarrow \quad 1 \leq 3 \leq 10$ ∎

Entrada

La entrada es una línea que contiene dos números S_1 y S_2, con $1 \leq S_1 < S_2 \leq 25000$.

Salida

La salida contiene todos los pares de *números afines* del rango indicado en la entrada. Cada línea posee un par de *números afines* N_i y M_i separados por un espacio, con $S_1 \leq N_i < M_i \leq S_2$ y $1 \leq i \leq T$. Además, deberán estar ordenados de forma ascendente $N_i < N_{i+1}$. En caso que no haya *números afines* (T = 0) la salida es una única línea con el valor 0.

Ejemplos

Entrada 1

```
9 87
```

Salida 1

```
10 14
12 16
15 16
18 21
20 22
40 44
40 50
56 64
```

Entrada 2

```
4000 5000
```

Salida 2

```
0
```

Credit for this problem: **Cristian Martinez**

Problema 6: Súper Ranking de Stark Games

SPOJ code: STARKG

La compañía de juegos en línea Stark Games realizará una competencia por equipos que otorgará premios para la Copa del Mundo de Qatar 2022, incluyen pasajes, estadías y entradas. Para determinar el/los equipos ganadores no se considera directamente la cantidad de victorias obtenidas por un equipo, sino que se confeccionarán dos rankings: ranking hora y súper ranking.

Los juegos están siempre habilitados y cada hora se contabilizan las victorias de los equipos de la siguiente forma: para elaborar el ranking hora se consideran las victorias de las últimas 10 horas (incluyendo la actual), es decir, en la hora H se contabilizan las victorias obtenidas en H, H-1, H-2,… H-9. Las victorias anteriores no son tenidas en cuenta. El/los equipos con mayor cantidad de victorias contabilizadas suman un punto en el súper ranking.

Cuando se cierra la competencia el/los equipos con mayor cantidad de puntos en el súper ranking ganan la competencia.

En total hay 26 equipos, cada uno identificado con una letra mayúscula de "A" a "Z", sin considerar "Ñ".

Entrada

La primera línea posee un único número S indica la cantidad de casos de prueba, $1 \leq S \leq 16$ y luego una línea en blanco. A continuación, para cada caso de prueba la primera línea posee un único número N que corresponde a la cantidad de horas con $1 \leq N \leq 2000$. Cada una de las siguientes N líneas contienen un 0, indicando que no hubo ningún equipo ganador en esa hora, o una lista de M letras, no ordenadas y posiblemente repetidas, separadas por un espacio (" "). Con $0 \leq M \leq 100$. Luego de cada caso de prueba hay una línea en blanco.

Salida

La salida posee dos líneas por cada caso de prueba. La primera línea es un único número con el primer puesto en puntos en el súper ranking. La segunda línea es una lista de M letras, con $M > 0$, indicando el/los equipos que obtuvieron los puntos de la línea anterior. En caso de haber más de un equipo se listan en orden alfabético.

Ejemplo

Entrada

```
4

2
Z A B Z A
W T T T A T

5
Y
D
0
0
Q D
```

```
12
A
B
0
0
0
0
0
0
0
0
0
0

10
0
0
0
0
0
T
P
0
0
0
```

Salida

```
1
A T Z
4
D Y
10
A B
5
T
```

Credit for this problem: **Cristian Martinez**

Problema 7: Carreras espaciales

SPOJ code: SPRACES

Es el año 3210 y tanto los viajes en el espacio como algunos mecanismos de teletransportación son algo muy normal. En esta época existe un nuevo deporte muy conocido, que se parece un poco a las carreras de autos de hoy en día.

Este nuevo deporte es una carrera de naves espaciales en línea recta, donde la pista tiene un cierto largo y cada nave corre a su propia máxima velocidad. Por cuestiones de seguridad, no se les permite a los corredores adelantar a otras naves, entonces cuando un corredor más rápido alcanza a otro más lento, el más rápido queda a la par del más lento hasta llegar al final de la pista.

Sin embargo, al final de la pista hay una máquina que teletransporta instantáneamente a las naves hacia el inicio de la pista, y entonces estas naves siguen corriendo. Cuando múltiples naves entran al teletransportador al mismo tiempo, todas aumentan su velocidad al máximo inmediatamente (y aquí entonces los corredores más rápidos adelantan a los más lentos).

Todos los corredores largan al mismo tiempo desde el inicio de la pista a su máxima velocidad. Es decir, que en el instante de la largada, los corredores más lentos no afectan a los más rápidos.

Se considera que una nave completa una vuelta cuando la misma es teletransportada desde el final de la pista, nuevamente hacia el inicio.

Dada la cantidad de vueltas que debe realizar cada corredor y la velocidad máxima de su nave medida en el tiempo que le lleva completar una vuelta, calcular en qué tiempo finalizará cada uno.

Entrada

La primera línea de entrada contiene un entero N ($1 \leq N \leq 5000$), indicando la cantidad de corredores. Luego siguen N líneas, donde la i-ésima de ellas contiene 2 enteros t_i y c_i ($1 \leq t_i \leq 10^6$, $1 \leq c_i \leq 1000$), que indican respectivamente el tiempo que le lleva completar una vuelta y la cantidad de vueltas que le faltan completar al i-ésimo corredor.

Los corredores están ordenados en orden decreciente según su velocidad. Es decir que $t_1 \leq t_2 \leq \ldots \leq t_N$ (recuerde que menor tiempo en dar una vuelta significa mayor velocidad).

Salida

Escriba N líneas en la salida, tal que la i-ésima línea contenga un entero indicando en qué tiempo completará la carrera el i-ésimo corredor.
Se puede demostrar que todos los tiempos de finalización serán enteros positivos.

Ejemplo

Entrada

```
2
4 8
7 6
```

Salida

```
36
42
```