

Streamer Stats Challenge

¡Los Premios ESLAND están a la vuelta de la esquina! TheGrefg y su equipo necesitan tu ayuda para analizar los datos de los streamers nominados y determinar los ganadores de varias categorías. Como sabes, los ESLAND son los premios más importantes del streaming en español, y este año han decidido que algunas categorías se decidirán usando datos reales de las retransmisiones.

Para hacer el análisis más interesante, no solo van a tener en cuenta el número de espectadores, sino también la interacción con el chat. Al fin y al cabo, ¿qué sería un stream sin su chat? Por eso, han estado recopilando dos métricas clave durante los últimos meses:

- El número de espectadores en cada momento
- La cantidad de mensajes en el chat

Con estos datos, quieren identificar no solo quién tiene más viewers, sino también qué streamers consiguen mantener a su chat más activo y quiénes son más consistentes con su audiencia. Por ejemplo, ¿sabías que a veces un stream con menos espectadores puede tener un chat mucho más activo?

Tu misión, si decides aceptarla, es crear un programa que analice estos datos y ayude a determinar:

- El momento más "hypeado" de cada streamer (cuando su chat estaba más activo en proporción a los viewers).
- Las mejores rachas de crecimiento de cada uno.
- Y por supuesto, ¡los ganadores de las diferentes categorías!:
 - **PREMIO AUDIENCIA:** Streamer con el pico más alto de viewers.
 - **PREMIO RATIO:** Streamer con la mejor ratio promedio de mensajes/viewers respecto a todas las mediciones.
 - **PREMIO CONSTANTE:** Streamer más "constante" (menor diferencia entre su máximo y mínimo de viewers).

Entrada

La entrada tendrá el siguiente formato:

1. La primera línea contiene un número N ($1 \leq N \leq 100$) que indica el número de streamers.
2. Las siguientes N líneas contienen:
 - El nombre del streamer (1 sola palabra)
 - El número M ($1 \leq M \leq 50$) de mediciones
 - Seguido de M pares de números enteros que representan en cada momento:
 - Número de espectadores
 - Número de mensajes en el chat

Salida

El programa debe calcular y mostrar:

1. Para cada streamer:
 - Su pico máximo de espectadores.
 - El "momento más hypeado" empezando a contar los datos desde 1, es decir, cuando la ratio mensajes/espectadores fue más alta, además hay que mostrar dicho ratio de mensajes/espectadores entre paréntesis.
 - La racha más larga donde los espectadores fueron aumentando.
2. Al final:
 - El streamer con el pico más alto de viewers
 - El streamer con la mejor ratio promedio de HYPE (mensajes/viewers)
 - El streamer más "constante" (menor diferencia entre su máximo y mínimo de viewers. Fórmula: $((\text{MAX}(\text{viewers}) - \text{MIN}(\text{viewers})) \times 100) / \text{MAX}(\text{viewers})$)

Formato de la Salida

Para cada streamer se mostrará:

[Nombre]
MAX= [pico]
HYPE= [n] ([ratio])
RACHA= [n]

Al final se mostrará:

PREMIOS ESLAND
PREMIO AUDIENCIA [nombre] [viewers]
PREMIO RATIO [nombre] [ratio]
PREMIO CONSTANTE [nombre] ([porcentaje])

Notas

- Los ratios y los porcentajes deben mostrarse con 2 decimales.
- En caso de empate, se premiará al streamer que aparezca primero en la entrada.
- Nunca se pasará como entrada un streamer sin datos.

Ejemplo 1: Entrada de multiples streamers y multiples medidas

Entrada:

3

Rubius 4 1000 500 1500 800 2000 1200 1800 1000

Ibai 3 2500 1500 3000 2000 2800 1800

Auronplay 5 1200 600 1300 700 1100 500 1400 800 1600 1000

Salida:

Rubius

MAX= 2000

HYPE= 3 (0.60)

RACHA= 3

Ibai

MAX= 3000

HYPE= 2 (0.67)

RACHA= 2

Auronplay

MAX= 1600

HYPE= 5 (0.62)

RACHA= 3

PREMIOS ESLAND

PREMIO AUDIENCIA Ibai (3000)

PREMIO RATIO Ibai (0.64)

PREMIO CONSTANTE Ibai (16.67)

Ejemplo 2: Entrada de un único streamer y multiples medidas

Entrada:
1 EIRubius 5 1000 100 1500 200 2000 300 2500 400 3000 500

Salida:
EIRubius MAX= 3000 HYPE= 5 (0.17) RACHA= 5 PREMIOS ESLAND PREMIO AUDIENCIA EIRubius (3000) PREMIO RATIO EIRubius (0.14) PREMIO CONSTANTE EIRubius (66.67)

Ejemplo 3: Entrada de un único streamer y multiples medidas con alta variación

Entrada:

1
Ibai 4 5000 1000 1000 800 8000 2000 2000 1000

Salida:

Ibai
MAX= 8000
HYPE= 2 (0.80)
RACHA= 2

PREMIOS ESLAND
PREMIO AUDIENCIA Ibai (8000)
PREMIO RATIO Ibai (0.44)
PREMIO CONSTANTE Ibai (87.50)

Puntuación

Test 1: Entrada de multiples streamers y multiples medidas (60 puntos)

Test 2: Entrada de un único streamer y multiples medidas (40 puntos)

Criterios Generales:

1. El formato de salida debe coincidir exactamente con el especificado
2. Los valores numéricos deben tener la precisión requerida
3. No se requiere validación de entrada

El emperador de 'words'

Andreu y Berto tienen un programa en una conocida cadena de radio. En él programa tienen un concurso que llaman 'words'. En el concurso Andreu propone una palabra y Berto tiene que proponerle otra que esté de algún modo relacionada con la palabra original.

Recientemente han ido a una entrevista a la tele, al programa de David, quien con malas artes le ha arrebatado a Berto el título de *emperador de 'words'*. Berto se ha negado a volver a jugar hasta que Andreu deje claras unas reglas para la valoración de las palabras propuestas por los concursantes.

Después de mucho discutir, han acordado las siguientes reglas:

- Por cada letra que esté presente en la palabra original: 1 punto.
- Por cada secuencia de dos letras que este presente en la palabra original: 2 puntos.
- Por cada secuencia de tres letras que esté presente en la palabra original: 3 puntos.
- Y así sucesivamente.
- Una palabra quedará descalificada, con una puntuación final de -1 si se ha copiado mucho de la palabra original, esto Andreu lo decidirá con un número, que indicará la longitud máxima de esa secuencia que puede coincidir entre palabras.
- Si la palabra propuesta es más corta que el número queda también descalificada.
- Cada secuencia solo puede contar una única vez.

A estas secuencias de n-elementos, les llamamos n-gramas, un 1-grama es una única letra, un 2-grama son dos letras, etc.

Siguiendo estas reglas:

- n=3 palabra_original=nevera, tetera estaría descalificada por tener un 3-grama en común con nevera.
- n=4 palabra_original=nevera, tetera obtendría 10 puntos (a, e, r), (ra, er), (era)

- $n=4$ palabra_original=nevera, palabra obtendría 4 puntos (a, r), (ra)

Esto es un juego competitivo, por tanto varios participantes pueden participar con sus palabras. Tu tarea es implementar un programa que permita juzgar las palabras propuestas, calcule los puntos de cada palabra, y muestre el ránking de las palabras ordenado por puntos.

Entrada:

La entrada consiste en una única línea de texto que tiene primero el número de Andreu, a continuación las palabra propuesta por Andreu, seguida por las palabras que concursan.

```
n palabra_original palabra_propuesta_1 palabra_propuesta_2 ...
```

Salida

Se procederá a imprimir en orden descendente de puntuación la palabra y su puntuación. En caso de empate, se debe de seguir el orden en el que hemos recibido las palabras

Ejemplos

Entrada

```
1 maneras nevera
```

Salida:

```
nevera -1
```

Entrada

```
1 nevera tetera
```

Salida:

```
tetera -1
```

Entrada

```
2 nevera horno
```

Salida:

```
horno 2
```

Entrada

3 nevera eva

Salida:

eva 5

Entrada

4 nevera tetera tartera

Salida:

tetera 10

tartera 10

Entrada

3 nevera tetera frigorifico

Salida:

frigorifico 1

tetera -1

Entrada

4 nevera tetera frigorifico

Salida:

tetera 10

frigorifico 1

Puntuación

- **Test 1 (20 puntos):** Calcula bien la puntuación de 1-gramas de 1 palabra
- **Test 2 (30 puntos):** Calcula bien la puntuación de 2-gramas de 1 palabra
- **Test 3 (50 puntos):** Calcula bien la puntuación de n-gramas ($n \leq 100$) de m palabras ($m \leq 100$)

Viajando con números en la cabeza

Cuando éramos pequeños, los viajes en coche se nos hacían interminables. Inventábamos mil juegos para hacer más llevaderas las horas y horas que pasábamos en el coche sin poder movernos.

Uno de los juegos al que más jugábamos era el de palabras encadenadas: El primer jugador dice una palabra cualquiera y el siguiente debe decir una palabra que comience con la última sílaba o letra de la palabra anterior. La idea es no repetir las palabras ya dichas.

En uno de los viajes especialmente largos, en el que al final se nos acabaron las palabras, decidimos cambiar el juego cambiando palabras por números: El primer jugador dice un número de n cifras (como mínimo 2 cifras y como máximo 10 cifras) y el siguiente debe decir un número también de n cifras, **incluyendo ceros a la izquierda si fuese necesario**, que comience con las $n/2$ últimas cifras del número anterior. **La idea es no repetir los números ya dichos.**

Si n (el número de cifras) es par, los números se parten por la mitad. Si el número es impar, el número debe contener como inicio la mitad redondeada al entero superior.

Así si por ejemplo el primer jugador dice 236, el segundo podría decir 360. A continuación podría decir 601. El siguiente 018. El siguiente 183, y así sucesivamente.

(Modo de funcionamiento 1 (Ver ENTRADA/SALIDA para más detalles))

El modo de funcionamiento 1 del programa permite determinar si una secuencia de un mínimo de 2 y un máximo de 100 elementos cumple con estos requisitos de juego.

Tras un rato jugando a este juego, nos dimos cuenta que era demasiado fácil y decidimos complicarlo ligeramente... Solo vamos a admitir en la secuencia números primos, sin repetirlos.

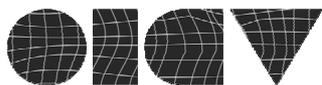
(Modo de funcionamiento 2 (Ver ENTRADA/SALIDA para más detalles))

El modo de funcionamiento 2 del programa permite determinar si una secuencia de un mínimo de 2 y un máximo de 100 elementos cumple con los nuevos requisitos del juego.

Ya en el destino, empecé a cavilar sobre el juego y me pregunté: ¿Cuál será la secuencia de primos que se puede formar escogiendo siempre los números primos más grandes sin repetir ninguno empezando por un número primo dado?

Así, si empezásemos por 19, el siguiente primo debería ser 97 (99 es múltiplo de 3 y 98 es múltiplo de 2). El siguiente debería ser 79. A partir de ahí ya no se podría seguir porque no queda ningún número primo que empiece por 9 (96, 94, 92 y 90 múltiplos de 2, 95 múltiplo de 5, 93 múltiplo de 3 y 91 múltiplo de 7).

O si empezásemos por 139, el siguiente sería 397 (399 es múltiplo de 3 y 398 es par). A continuación iría el 977 (999 es múltiplo de 3 y 998 es par). Luego iría 773 (779 es divisible por 19, 778, 776 y 774 son pares, 777 es múltiplo de 3 y 775 es múltiplo de 5). Después tendríamos el 739 y aquí acabaría la secuencia, ya que ya hemos puesto el



397 y 396, 394 y 392 son pares, 395 y 390 son múltiplos de 5, 393 es múltiplo de 3 y 391 es múltiplo de 17.

(Modo de funcionamiento 3 (Ver ENTRADA/SALIDA para más detalles))

El modo de funcionamiento 3 del programa nos dirá cuál es la longitud y la secuencia formada con los mayores primos que se puedan ir añadiendo partiendo de un número primo dado.

ENTRADA y SALIDA

El programa tendrá tres modos de funcionamiento, indicados como 1, 2 o 3 en la primera línea de entrada. Si fuese el modo 1 o 2, a continuación en la siguiente línea, el número de secuencias (a lo sumo 10 secuencias) a comprobar y, en líneas separadas, las secuencias a comprobar: tamaño de la secuencia (un mínimo de 2 y un máximo de 100 elementos) y los elementos de la secuencia separados por espacios en blanco. Si fuese el modo 3, en la línea siguiente el número de secuencias a obtener (a lo sumo 10) y en líneas separadas los números generadores de dichas secuencias. El número generador NO podrá tener ceros a la izquierda.

En los modos 1 y 2, si la secuencia es correcta la salida será un 1. Si la salida es incorrecta se mostrará un 0.

En el modo 3 se mostrará como salida la longitud de la secuencia seguida de la secuencia generada. La información de cada secuencia en una línea separada. Si el número generador no fuese un número primo, la longitud de la secuencia generada sería 0 (y, evidentemente, no habría elementos en la secuencia).

En todos los casos los enteros de la secuencia serán valores positivos mayores o iguales a 10 y menores o iguales a 2000000 y los números que inician las secuencias nunca tendrán ceros a la izquierda. Con estas condiciones, las secuencias de primos encadenados son de, como máximo, 20000 elementos.

Ejemplos de entradas y salidas:

Ejemplo 1: Modo de funcionamiento 1

Entrada	Eixida
1	1
6	0
4 236 360 601 018	1
4 236 360 601 18	0
5 9826 2671 7182 8254 5479	0
4 223 231 305 153	0
7 9826 2671 7182 8254 5426 2671 7119	
4 9826 2671 718 1821	

La secuencia 2 es incorrecta porque uno de los números no tiene las cifras correctas (3 del número original: Al 18 le falta el cero a la izquierda)

La secuencia 4 es incorrecta porque el tercer número no empieza con 31

La secuencia 5 es incorrecta porque se repite el número 2671

La secuencia 6 es incorrecta porque uno de los números no tiene las cifras correctas (4 del número original)

Ejemplo 2: Modo de funcionamiento 2

Entrada	Eixida
2	1
5	0
4 223 233 337 379	0
4 223 239 937 373	0
7 9829 2939 3929 2939 3911 1103 0311	0
5 3929 2939 3911 1103 311	
4 9829 2939 3939 3929	

La secuencia 2 no es correcta porque el tercer número de la secuencia no empieza por 39

La secuencia 3 no es correcta porque se repite el número 2939

La secuencia 4 es incorrecta porque uno de los números no tiene las cifras correctas (4 del número original: Al 311 le falta el cero a la izquierda)

La secuencia 5 no es correcta porque el número 3939 no es primo

Ejemplo 3: Modo de funcionamiento 3

Entrada	Salida
3	7 13 37 79 97 73 31 19
3	11 701 019 199 997 977 773 739 397 971 719 197
13	0
701	
1234	

Puntuación:

Test 1 (15 puntos): Modo de funcionamiento 1.

Test 2 (25 puntos): Modo de funcionamiento 2.

Test 3 (60 puntos): Modo de funcionamiento 3.

SUDOKU CHECKER

Sudoku és un passatemps que consisteix a completar amb números de l'1 al 9 una quadrícula de 81 caselles (9 files per 9 columnes) dividida en 9 blocs de 3x3 caselles, on alguns números ja apareixen col·locats a la quadrícula a manera de pistes, de manera que no es repeteixi cap número a la mateixa fila o columna ni a la mateixa subquadrícula.

A tall d'exemple, aquí tens una solució *sudoku* correcta:

9	2	3	7	6	1	8	4	5
5	4	6	8	9	2	3	7	1
8	1	7	5	4	3	9	6	2
7	6	9	2	5	4	1	3	8
1	3	5	9	8	6	4	2	7
4	8	2	1	3	7	6	5	9
2	7	4	3	1	9	5	8	6
6	9	8	4	7	5	2	1	3
3	5	1	6	2	8	7	9	4

Has d'implementar un programa que comprovi si una solució *sudoku*, que pot contenir cel·les buides, és vàlida o no. Una solució *sudoku* serà una matriu de 9x9 números del 0 al 9, on el 0 indica una casella buida.

Per considerar una solució com a vàlida:

- No pot contenir caselles buides (zeros)
- No hi pot haver dos o més dígitos de l'1 al 9 iguals a una mateixa fila, columna o bloc de 3x3 caselles.

ENTRADA

El programa tindrà dos modes de funcionament, indicats com 1 o 2 a la primera línia d'entrada (més informació a l'apartat SORTIDA). Tot seguit apareixen 9 línies de text, totes amb salt de línia al final, amb nou dígitos entre 0 i 9 separats per espais, on 0 indica una casella buida.

Dos exemples d'entrada:

Solució no vàlida	Solució vàlida (la de l'exemple de dalt)
<p>2</p> <p>1 2 3 4 5 6 7 8 9</p> <p>2 0 4 5 6 7 8 9 1</p> <p>3 4 5 0 7 8 9 1 2</p> <p>4 5 6 7 8 9 1 2 3</p> <p>5 6 7 8 9 1 9 3 4</p> <p>6 7 8 9 1 0 3 4 5</p> <p>7 8 9 1 2 3 4 1 6</p> <p>8 9 1 2 3 4 5 6 7</p> <p>9 1 2 3 4 5 6 7 8</p>	<p>1</p> <p>9 2 3 7 6 1 8 4 5</p> <p>5 4 6 8 9 2 3 7 1</p> <p>8 1 7 5 4 3 9 6 2</p> <p>7 6 9 2 5 4 1 3 8</p> <p>1 3 5 9 8 6 4 2 7</p> <p>4 8 2 1 3 7 6 5 9</p> <p>2 7 4 3 1 9 5 8 6</p> <p>6 9 8 4 7 5 2 1 3</p> <p>3 5 1 6 2 8 7 9 4</p>

SORTIDA

Mode de funcionament 1:

- S'imprimeix OK si la solució és vàlida i ERROR si no ho és.

Mode de funcionament 2:

- S'imprimeix OK si la solució és vàlida i ERROR si no ho és.
- A més, si la solució no és vàlida, imprimeix una quadrícula revelant les cel·les que invaliden el *sudoku*. Les cel·les que no estiguin buides i continguin números no duplicats a la mateixa fila, columna o bloc, apareixeran com un punt. Per a l'exemple d'entrada 'solució no vàlida' de dalt, la sortida seria

ERROR

```
. 2 3 . 5 6 . 8 9
2 0 4 5 6 7 8 9 1
3 4 . 0 7 . 9 1 .
. 5 6 . 8 9 . . 3
5 6 7 8 9 1 9 3 4
6 7 . 9 1 0 3 4 .
. 8 9 1 2 3 . 1 6
8 9 1 2 3 4 . 6 7
9 1 . 3 4 . 6 7 .
```

PUNTUACIÓ



IV Olimpiada d'Informàtica Comunitat Valenciana

Universitat d'Alacant - Escola Politècnica Superior

Universitat de València - Escola Tècnica Superior d'Enginyeria

- **Test 1 (30 punts):** Mode de funcionament 1.
- **Test 2 (70 punts):** Mode de funcionament 2.