

Nivel 2do Año
Problem Set

3 de Noviembre de 2023

Problema 1: Anagramas Numéricos

SPOJ code: TM23AN

Tiempo límite: 1 segundo.

Un anagrama numérico es un cambio en el orden de los dígitos de un número que da lugar a otro número distinto. Por ejemplo, los anagramas de N = 105 son: 015, 051, 150, 501 y 510.

Luego, a partir del conjunto que componen N y sus anagramas puede obtenerse la mayor diferencia entre cualquier par de números del conjunto. Para el ejemplo anterior la mayor diferencia es 495 (510 - 15).

Entrada

Un número N, con $0 \leq N \leq 2^{1024}$.

Salida

Un número M ($0 \leq M < 2^{1027}$) que corresponde a la mayor diferencia entre dos números cualesquiera del conjunto que componen la entrada (N) y sus anagramas.

Ejemplos

Entrada 1

```
1000
```

Salida 1

```
999
```

Entrada 2

```
444444
```

Salida 2

```
0
```

Credit for this problem: **Cristian Martínez**

Problema 2: Las Secuencias de Marina

SPOJ code: TM23LSDM

Tiempo límite: 4 segundos.

En sus tiempos libres, Marina inventa juegos matemáticos para entretenerse con sus amigas. Su última creación consiste en sumar números adyacentes de una secuencia.

Se comienza con un número *N* sin ceros no significativos a izquierda. Luego, los *m* dígitos de *N* se consideran como una secuencia de números N_1 N_2 … N_m. A partir de allí, se suman los adyacentes obteniendo una nueva secuencia. El paso anterior se repite hasta que la secuencia obtenida tenga un único número. Por ejemplo, *N* = 2027:

Orden	Secuencia
1°	**2 0 2 7**
2°	**2 2 9**
3°	**4 11**
4°	**15**

Entrada

La primera línea posee un único número *N*, con $10 \leq N \leq 2^{1024}$.

Salida

La salida serán *M* líneas con una secuencia de números cada una de ellas, y luego una línea en blanco. Cada número en la secuencia es seguido por un espacio (' '). La primera línea corresponde a la entrada expresada como una secuencia de dígitos. A partir de la segunda línea, la secuencia corresponde a la suma de los adyacentes de la secuencia previa.

Ejemplos

Entrada 1

```
45
```

Salida 1

```
4 5
9
```

Entrada 2

```
1111
```

Salida 2

```
1 1 1 1
2 2 2
4 4
8
```

Credit for this problem: **Cristian Martínez**

Problema 3: Amistad

SPOJ code: TM23AMI

Tiempo límite: 1 segundo.

En algún tranquilo y pacífico prado de Lisse en los Países Bajos, *Bram* se encuentra inmerso en un cálido cortejo con todos los amigos que ha abrazado a lo largo de su vida. El calor de la fogata sobre la que reposan dota de ligereza al sereno, pero a la vez amigable ambiente. Después de tantas horas de diversión y jocosidad, *Bram* decide que es el momento idóneo para poner a prueba la afinidad del grupo proponiéndole a sus invitados intentar formar un *Círculo de la Amistad*.

La creación de un *Círculo de la Amistad* implica colocar un tronco por cada invitado alrededor de la fogata tal que éstos formen un círculo y en donde cada invitado elegirá dónde sentarse. Luego *Bram*, quien se encuentra con los ojos vendados, dirá el nombre de cada uno de los invitados, quienes, a medida que van siendo nombrados, abandonan el círculo.

Se dice que un grupo sortea la prueba del *Círculo de la Amistad* si cada vez que un invitado abandona el círculo, el siguiente en hacerlo es aquel que se encontraba inmediatamente a la derecha de este. Su tarea consiste en proporcionar a *Bram* un programa que le permita determinar si sus amigos superaron la prueba dado un vector circular ordenado de la misma forma en la que estaban dispuestos en el momento en el que cada uno de ellos abandonó el círculo.

Entrada

La entrada consta de múltiples líneas. La primera línea contiene un entero $1 \leq N \leq 10^5$, que representa la cantidad de invitados. La siguiente línea contiene N números enteros $1 \leq M_i \leq N$, representando el orden en el que el invitado ubicado en la posición i abandona el círculo.

Salida

La salida consta de una única línea, sin salto de línea al final, con la palabra “SI” en caso de que la prueba sea superada y “NO” en caso contrario.

Ejemplo

Entrada 1

```
10
7 8 9 10 1 2 3 4 5 6
```

Salida 1

```
SI
```

Entrada 2

```
7
1 2 3 4 6 5 7
```

Salida 2

```
NO
```

Credit for this problem: **Lucio Rivera**

Problema 4: Mensajes Seguros

SPOJ code: TM23MS

Tiempo límite: 1 segundo.

Una organización detectó que las comunicaciones internas están siendo interceptadas y modificadas por una persona infiel que aún no pudo ser descubierta.

Como medida de seguridad se decidió agregar a cada mensaje un verificador. El verificador es un número generado por el emisor a partir del mensaje al momento de enviarlo. Cuando el mensaje llega al receptor, éste constata si el número verificador coincide con el enviado.

La forma de calcular el verificador consiste en obtener el valor ascii de cada uno de los caracteres que componen el mensaje y multiplicarlo por la posición que ocupa en el mensaje, obteniendo así un valor N_k. Luego, la suma de todos los valores N_k es el verificador. Por ejemplo, para el mensaje 'TOP SECRET.':

Caracter	**T**	**O**	**P**		**S**	**E**	**C**	**R**	**E**	**T**	**.**
Valor ascii	84	79	80	32	83	69	67	82	69	84	46
Posición	1	2	3	4	5	6	7	8	9	10	11
N_k	84	158	240	128	415	414	469	656	621	840	506

El mensaje 'TOP SECRET.' tiene 11 caracteres, su verificador es $\sum_{k=1}^{11} N_k = 4531$.

En la Tabla 1 se encuentran los caracteres que pueden ser incluidos en los mensajes junto a sus valores ascii.

Caracteres ASCII					
32	espacio	65	A	78	N
44	,	66	B	79	O
46	.	67	C	80	P
48	0	68	D	81	Q
49	1	69	E	82	R
50	2	70	F	83	S
51	3	71	G	84	T
52	4	72	H	85	U
53	5	73	I	86	V
54	6	74	J	87	W
55	7	75	K	88	X
56	8	76	L	89	Y
57	9	77	M	90	Z

Tabla 1. Caracteres y sus valores ascii.

Entrada

La primera línea consiste en una secuencia de caracteres S con $0 \leq \text{longitud}(S) \leq 20000$. S contiene únicamente los caracteres: punto ('.'), coma (','), letras mayúsculas ('A'-'Z') y números ('0'-'9'). No se incluyen Ñ ni caracteres acentuados. La segunda línea tiene un único número N con $0 \leq N < 2^{35}$. N representa el número verificador.

Salida

La salida es una única línea con una de las siguientes alternativas:

- Si el número verificador corresponde al mensaje: un string 'ACEPTADO'.
- Si el número verificador no corresponde al mensaje: un string 'RECHAZADO', seguido por un espacio (' ') y luego el número verificador correspondiente al mensaje.

Ejemplos

Entrada 1

```
TOP SECRET.
4531
```

Salida 1

```
ACEPTADO
```

Entrada 2

```
COMPETENCIA DE PROGRAMACION TECNOMATE 2023
1000
```

Salida 2

```
RECHAZADO 59056
```

Entrada 3

```
A B C
795
```

Salida 3

```
RECHAZADO 790
```

Credit for this problem: **Cristian Martínez**

Problema 5: Lluvia de Meteoritos

SPOJ code: TM23LDM

SpaceY es una compañía dedicada a la investigación de las emisiones de radioactividad generadas por las lluvias de meteoritos. Para ello, realiza simulaciones de impactos de meteoritos en distintos puntos de una región.

La simulación incluye:

- Región: puede ser representada como una cuadrícula de tamaño N x M. Cada celda tiene asignada una posición (I, J) y un valor W que indica la carga radiactiva de la misma. Al comienzo la carga radiactiva de todas las celdas es cero.
- Meteorito: del cual se conoce lugar de impacto (una celda) y su radiactividad.

Tener en cuenta:

1. Con el paso del tiempo la radiactividad se reduce. Al comenzar un intervalo de tiempo la carga radiactiva de toda la región se reduce en una unidad (nunca puede ser negativa).
2. Durante cada intervalo de tiempo puede ocurrir a lo sumo un impacto.
3. El impacto de un meteorito con radiactividad W' incrementa la carga radiactiva W de la celda, es decir, $W = W + W'$.
4. Una celda puede recibir varios impactos en distintos intervalos.

El objetivo de SpaceY es determinar la cantidad de celdas afectadas luego de la simulación.

Entrada

La primera línea posee tres números N, M, T: N es la cantidad de filas de la cuadrícula ($1 \leq N \leq 500$), M es la cantidad de columnas de la cuadrícula ($1 \leq M \leq 500$), y T corresponde a la cantidad de intervalos de tiempo de la simulación ($1 \leq T \leq 1000$).

Cada una de las siguientes T líneas contienen tres números I, J y W, cada uno de ellos separados por espacio (' '). Con $0 \leq I \leq 500$, $0 \leq J \leq 500$ y $0 \leq W \leq 2000$. Siendo (I, J) la celda del lugar del impacto y W la radioactividad. En el caso de que no hubiere impacto en un intervalo de tiempo, los tres números serán cero.

Salida

La salida es una única línea con un número N, con $0 \leq N \leq 250000$, indicando la cantidad de celdas con la carga radiactiva mayor a cero.

Ejemplos

Entrada 1

```
10 6 6
0 0 0
2 1 15
2 1 900
4 2 1159
6 4 90
2 4 20
```

Salida 1

```
4
```

Entrada 2

```
200 300 3
24 291 2
199 280 2
0 0 0
```

Salida 2

```
1
```

Entrada 3

```
500 499 4
107 312 3
0 0 0
0 0 0
0 0 0
```

Salida 3

```
0
```

Credit for this problem: **Cristian Martínez**

Problema 6: Fortday

SPOJ code: TM23FDAY

Tiempo límite: 1 segundo.

Fortday es un juego en línea en el que participan jugadores de forma individual y grupal. Una partida consiste de dos contrincantes (o equipos) que se integran momentos antes de la misma.

Un equipo puede tener entre 1 y 10 jugadores como máximo. Además, los contrincantes que se enfrentan no necesariamente tienen la misma cantidad de integrantes.

Un jugador se identifica con una letra mayúscula o minúscula (excepto ñ y Ñ), por lo que hay un máximo de 52 jugadores (26 con letras A-Z y 26 con letras a-z). Un equipo es la secuencia de letras de sus integrantes.

Luego de cada partida los puntos obtenidos por un equipo se multiplican por la cantidad de jugadores del equipo contrincante. Luego, este puntaje se asigna a cada uno de los integrantes. Por ejemplo:

- Partida 1: **GhWP** 50 puntos – **qA** 40 puntos
- Partida 2: **W** 14 puntos – **qZA** 16 puntos

En la partida 1, cada uno de los jugadores **G**, **h**, **W** y **P** obtuvo 100 (50*2) puntos, mientras que cada uno de los jugadores **q** y **A** obtuvo 160 (40*4) puntos. En la partida 2, el jugador **W** obtuvo 42 (14*3) puntos, mientras que los jugadores **q**, **Z** y **A** sumaron 16 puntos. (Las partidas 1 y 2 corresponden al ejemplo Entrada 1).

Los puntos de cada partida son acumulativos. El premio es para el jugador que mayor cantidad de puntos obtuvo. En caso de empate, el premio es para todos los jugadores con el mayor puntaje.

Entrada

La primera línea posee un único número N, con $1 \leq N \leq 10000$, indicando la cantidad de partidas. Cada una de las siguientes N líneas contienen: un string S_1, seguido por un espacio (' '), seguido por un número P_1, seguido por un espacio (' '), seguido por un string S_2, seguido por un espacio (' ') y finalmente un número P_2. Con:

- $1 \leq \text{longitud}(S_1) \leq 10$ y $1 \leq \text{longitud}(S_2) \leq 10$.
- $0 \leq P_1 \leq 50000$ y $0 \leq P_2 \leq 50000$.
- S_1 y S_2 contienen únicamente caracteres 'A'-'Z' y 'a'-'z'.
- S_1 y S_2 no necesariamente están ordenados.

Cada una de las N líneas indica una partida: S_1 y S_2 incluyen las letras de los jugadores, y P_1 y P_2 los puntos obtenidos por S_1 y S_2, respectivamente.

Salida

La salida es una única línea con M letras, $1 \leq M \leq 52$, sin separación entre ellas, indicando el/los jugadores premiados individualmente ordenados lexicográficamente (primero mayúsculas A-Z y luego minúsculas a-z).

Ejemplos

Entrada 1

```
2
GhWP 50 qA 40
W 14 qZA 16
```

Salida 1

```
Aq
```

Entrada 2

```
2
WaZ 10 j 6
Z 7 a 6
```

Salida 2

```
j
```

Entrada 3

```
1
caB 0 qQ 0
```

Salida 3

```
BQacq
```

Credit for this problem: **Cristian Martínez**

Problema 7: Misteriosas Montañas del Sur

SPOJ code: TM23MMS

Tiempo límite: 1 segundo.

Climby es un aventurero que disfruta pasar tiempo recorriendo y explorando las Misteriosas Montañas del Sur. Una región que adquirió su nombre debido a la presencia de portales que teletransportan a quienes los cruzan.

Nuestro aventurero inicia el recorrido con cierta energía. En la medida que avanza por el paisaje, siempre un paso por vez, se cansa y únicamente en los descensos puede recuperarse:

- Un paso en ascenso: pierde 2 unidades de energía.
- Un paso en camino recto: pierde 1 unidad de energía.
- Un paso en descenso: gana 3 unidades de energía.

Climby no siempre se prepara lo suficiente y puede agotarse durante el recorrido. Si no tuviese energía suficiente para avanzar queda varado en ese lugar. Dado que en descenso gana energía, hay dos posibilidades: a) si está en camino recto, envía un pedido de RESCATE; b) si está en ascenso, envía un pedido de AUXILIO.

Cuando Climby llega al final del recorrido, aunque en el último esfuerzo se quedase sin energía, alcanzó a completarlo.

Consideraciones sobre los portales:

- Quien cruza un portal se teletransporta al portal inmediato posterior.
- Los portales no modifican la energía de quien lo cruza.
- La cantidad de portales en un recorrido siempre es par.

Entrada

La primera línea posee un número N, con $1 \leq N \leq 1000$, indicando las unidades de energía de Climby al inicio del recorrido. La segunda línea es un string S, con $1 \leq$ longitud(S) ≤ 10000. S está compuesto por los siguientes caracteres:

- '-' (guión) indica paso en camino recto.
- '/' (barra) indica paso en ascenso.
- '\' (barra invertida) indica paso en descenso.
- '=' (igual) indica un portal.

Salida

La salida es una única línea con un string S seguido por un espacio (' '), y a continuación un número N ($0 \leq N \leq 31000$) que indica las unidades de energía de Climby al detener su avance. El valor de S es una de las siguientes alternativas:

- 'COMPLETO', si Climby alcanzó a completar el recorrido.
- 'RESCATE', si Climby no puede avanzar estando en un paso recto.
- 'AUXILIO', si Climby no puede avanzar estando en un paso en ascenso.

Ejemplos

Entrada 1

```
10
--\\--/\\-
```

Salida 1

```
COMPLETO 15
```

Entrada 2

```
6
-//=-\\\=/
```

Salida 2

```
AUXILIO 1
```

Entrada 3

```
20
=\\\\--////=
```

Salida 3

```
COMPLETO 20
```

Entrada 4

```
7
//---
```

Salida 4

```
COMPLETO 0
```

Credit for this problem: **Cristian Martínez**

Problema 8: Tesoro Maya

SPOJ code: TESORO

Tiempo límite: 1 segundo.

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 ≤ 10^9 (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 Martínez**

Problema Adicional

Problema 9: Querido Watson

SPOJ code: TM23QW

El doctor Watson, cansado de ser la sombra del gran Sherlock Holmes, decidió dedicarse en su totalidad al estudio de bacterias y microorganismos. Descubrió que existe un tipo de bacteria que se agrupa formando una fila e interacciona de una manera muy interesante. Con el fin de facilitar su estudio, el doctor Watson enumeró a cada bacteria partiendo desde el uno y las clasificó dependiendo de qué tipo de bacteria eran.

Para poder continuar con su estudio, Watson necesita poder calcular de forma rápida la cantidad de bacterias distintas que existen en un intervalo. ¿Puedes ayudar a Watson a avanzar con su estudio?

Entrada

La entrada consta de múltiples líneas. La primer línea contiene dos enteros $1 \leq n \leq 2 \cdot 10^5$ y $1 \leq q \leq 2 \cdot 10^5$, la cantidad de bacterias en la fila, y la cantidad de consultas realizadas por Watson. La segunda línea contiene n enteros $1 \leq t_i \leq 10^9$, representando el tipo de bacteria en la posición i. Finalmente, le siguen q líneas con dos enteros $1 \leq a \leq b \leq n$, representando la consulta de Watson en el rango $[a, b]$.

Salida

Por cada consulta, deberá imprimir la cantidad de bacterias diferentes que existen en el rango [a, b], seguido de un salto de línea.

Ejemplo

Entrada

```
5 3
3 2 3 1 2
1 3
2 4
1 5
```

Salida

```
2
3
3
```

Credit for this problem: **Fernando Sebastián Lomazzi**