

Problema A. Amistad

Autor: *Lucio Rivera*
Tiempo límite: 1 segundo

En algún tranquilo y pacífico prado de Lisse en los Países Bajos, *Bram* se encuentra inmerso en un cálido encuentro con todos los amigos que ha cosechado a lo largo de su vida. El calor de la fogata que los ilumina 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 fogota tal que estos formen un círculo y en donde cada invitado elegirá donde 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 primer 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 línea, la palabra “SI” en caso de que la prueba sea superada y “NO” en caso contrario.

Ejemplos

| entrada estándar | salida estándar |
|----------------------------|-----------------|
| 10 7 8 9 10 1 2 3 4 5 6 | SI |
| 7 1 2 3 4 6 5 7 | NO |

Problema B. Bug en la Matrix

Autor: *Fernando Sebastián Lomazzi*
Tiempo límite: 1 segundo

Julián inventó un artefacto revolucionario, el *SIMP* (*Sistema Ideal de Multivisión Profunda*). Este artefacto consiste en unas gafas de realidad aumentada las cuales permiten ver el código genético subyacente de una persona, el cual determina características intrínsecas de la misma, nombrado ahora como la *Matrix*. Julián notó que cuando una persona poseía algún tipo de enfermedad, este código poseía un patrón particular, se encontraba infectado de *bugs*.

El código genético de una persona está determinado por una matriz de tamaño $n \times m$ (n filas y m columnas), compuesta por letras mayúsculas y minúsculas del alfabeto inglés que representan características de la persona. El nivel de enfermedad de una persona se mide de acuerdo a la cantidad de veces que aparece la palabra *bug*, en forma horizontal o vertical en cualquiera de los cuatro sentidos, sin diferenciar entre mayúsculas y minúsculas en su código genético.

Esta característica permitiría determinar enfermedades no diagnosticadas de las personas, por lo que es importante poder, en base al código genético de una persona, encontrar su nivel de enfermedad. Ayuda a Julián a poder determinar el nivel de enfermedad de las personas.

Entrada

La entrada consta de múltiples líneas. La primer línea contiene dos enteros $1 \leq n, m \leq 10^3$, la cantidad de filas y columnas de la matriz genotípica. Las siguientes n líneas contienen m letras del alfabeto inglés, en mayúsculas o minúsculas, representando el código genético de la persona estudiada.

Salida

La salida consta de una línea, el nivel de enfermedad de la persona en base a su código genético.

Ejemplos

| entrada estándar | salida estándar |
|---|-----------------|
| 4 5 bgbzb oujau gbgsg cvbbd | 2 |
| 4 3 bug ugb gbu bug | 4 |

Problema C. Contraseña Segura

Autor: *Lucio Rivera*
Tiempo límite: 1 segundo

La organización de *Tecnomate* le ha notificado a los autores de problemas de esta edición que para próximos años va a ser necesario que cada alumno posea una contraseña segura para darse de alta en el sistema de la competición.

Es bien sabido que los autores de esta edición son expertos en optimización de procesos siempre que esto implique menos horas de trabajo y más horas de ocio, ¡Cómo olvidar aquella vez que le pidieron a una Inteligencia Artificial que generase una competencia de programación!

Dado que aquí han identificado una oportunidad única de hacer uso de sus habilidades de delegación de trabajo, han decidido que los competidores de esta edición serán los encargados de desarrollar la pieza de software que verificará la seguridad de las contraseñas de los participantes en futuras ediciones.

La organización ha establecido que una contraseña s de tamaño n se considerará segura cuando la totalidad de las condiciones siguientes sean satisfechas:

- $5 \leq n \leq 32$.
- n debe ser un número impar.
- El caracter central debe ser un número primo.
- El primer caracter debe ser una letra minúscula.
- El último caracter debe ser una letra mayúscula.

¡Si no resuelves este problema no habrá una nueva edición de Tecnomate!

Entrada

La entrada consta de múltiples líneas. La primer línea contiene un entero $1 \leq t \leq 100$, que representa la cantidad de casos de prueba. Cada caso consiste en dos líneas, la primer línea contiene un entero $1 \leq n \leq 10^6$, la longitud de la contraseña s ; la segunda línea contiene la contraseña, compuesta de dígitos del 0 al 9, así como también letras mayúsculas y minúsculas del alfabeto inglés.

Se garantiza que la suma de los tamaños de las contraseñas entre todos los casos de prueba no supera 10^6 .

Salida

Para cada caso de prueba, deberá imprimir la frase “CONTRASENA SEGURA” si la contraseña es segura y “CONTRASENA NO SEGURA” en caso contrario.

Ejemplo

| entrada estándar | salida estándar |
|------------------|----------------------|
| 3 | CONTRASENA NO SEGURA |
| 11 | CONTRASENA SEGURA |
| sonMu1VagoS | CONTRASENA NO SEGURA |
| 7 | |
| xms7abC | |
| 14 | |
| demasiadoVagos | |

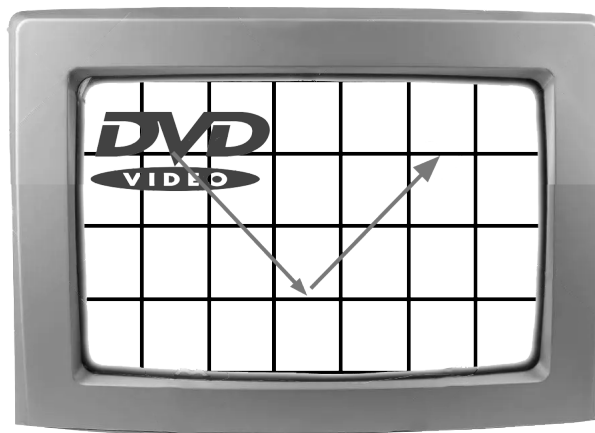
Problema D. DVD

Autor: *Fernando Sebastián Lomazzi*
Tiempo límite: 1 segundo

Lucas se encuentra viendo la nueva película *Chrek*, acerca de un orco verde y su asno en su reproductor de DVD. En un momento, *Chrek* y *Asno* son emboscados en una trampa planeada por el malvado *Lord Closequaad*, lo que enfureció tanto a *Lucas* que apagó la película y fue a por refrigerios.

A su vuelta, quedó fascinado ante el logo de DVD que se desplazaba en su pantalla. El mismo rebotaba constantemente, hasta que en un momento, chocó con la esquina superior izquierda. Luego de este increíble suceso, *Lucas* se preguntó: ¿Cuándo volverá el logo a golpear otra esquina de la pantalla?

Lucas tomó su regla y midió: su pantalla tiene 4 pulgadas de alto y 7 de ancho, mientras que el logo de DVD tiene 2 pulgadas de alto y 3 de ancho. Notó además que el logo se desplaza una pulgada vertical y horizontalmente cada segundo, y en caso de choque con un borde de la pantalla, cambia su sentido en la dirección de choque.



Lucas le solicita que realice un programa que calcule el tiempo que debe esperar considerando como condición inicial al logo en la esquina superior izquierda, con movimiento en sentido abajo-derecha. Deberá realizarlo para distintas mediciones, ya que desea ayudar a todo amante del cine con su misma pregunta.

Entrada

La entrada consta de múltiples líneas. La primera línea contiene un entero $1 \leq t \leq 10^4$, que representa la cantidad de casos de prueba. Cada caso consiste en dos líneas. La primera línea contiene dos enteros $2 \leq h, w \leq 10^9$, el alto y ancho de la pantalla en pulgadas. La segunda línea contiene dos enteros $1 \leq n < h$ y $1 \leq m < w$, el alto y ancho del logo de DVD en pulgadas.

Salida

Por cada caso de prueba, deberá imprimir la cantidad de segundos que deben transcurrir para que el logo de DVD choque, por primera vez, cualquier esquina de la pantalla partiendo desde la esquina superior izquierda.

Ejemplo

| entrada estándar | salida estándar |
|------------------|-----------------|
| 2 | 4 |
| 4 7 | 36 |
| 2 3 | |
| 6 13 | |
| 2 4 | |

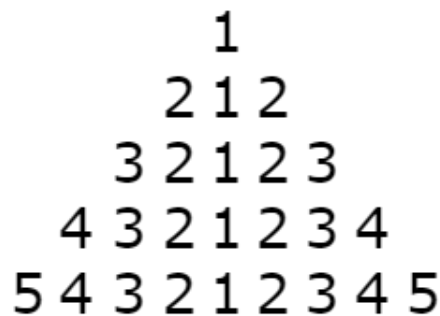
Problema E. El computador de la diosa Pitón

Autor: *Lucio Rivera*
Tiempo límite: 1 segundo

Un grupo de investigadores aficionados, bajo la dirección del renombrado experto *Roberto Martilloabierto* (conocido también como *Robert Oppenheimer* en inglés), han puesto en jaque el título que el *EINAC* (*Computador e Integrador Numérico Electrónico*) ostenta como el primer computador de propósito general de la historia de la humanidad. La evidencia sobre la que los investigadores se apoyan se basa en antiguos registros de una civilización poco conocida que floreció en el año 300 a.C., autodenominada *Los Desarrolladores de la diosa Pitón* (conocidos en inglés como *Python Devs*).

Entre las ruinas de la civilización, se encuentran plasmados enigmáticos jeroglíficos con forma de pirámide que Roberto, fervientemente cree, son en realidad registros de memorias utilizadas por los computadores primitivos, los primeros de la historia de la humanidad. El obstáculo inminente que Roberto debe sortear en el corto plazo es calcular muy rápidamente, la suma de los elementos que conforman estas pirámides.

Las pirámides de distintas alturas, aunque pudo observarse que han sido elaboradas siguiendo un mismo patrón entre sí.



Pirámide de altura 5.

Nótese que la columna central, de altura m , está compuesta por m elementos con valor 1, mientras que las dos columnas adyacentes constan de $m-1$ elementos con valor 2, las siguientes constan de $m-2$ elementos de valor 3 y así sucesivamente hasta llegar a columnas de los extremos, cada una de ellas compuestas por un solo elemento con valor m .

Usted, quien casualmente se encontraba en la zona sin mayores preocupaciones, ha sido reclutado a la fuerza por Roberto para colaborar en su investigación. Él percibe en usted una afinidad especial por las matemáticas y la computación, exactamente las habilidades necesarias para demostrar su teoría.

Entrada

La entrada consta de múltiples líneas. La primer línea contiene un entero $1 \leq t \leq 1000$, que representa la cantidad de casos de prueba. Cada caso consiste en un entero $1 \leq m \leq 10^6$, indicando la altura de la pirámide descrita.

Salida

Por cada caso de prueba, deberá imprimir la suma de los elementos de la pirámide.

Ejemplo

| entrada estándar | salida estándar |
|------------------|-----------------|
| 3 | 65 |
| 5 | 36 |
| 4 | 343300 |
| 100 | |

Nota

Roberto cree que quizás, la siguiente formula le pueda ser útil.

$$1^2 + 2^2 + \dots + k^2 = \frac{k(k+1)(2k+1)}{6}$$

Problema F. Frutigran

Autor: *Fernando Sebastián Lomazzi*
Tiempo límite: 1 segundo

Tomás y Mariano se juntaron a merendar para festejar el fin del cuatrimestre. Ambos llevaron su comida favorita para esta ocasión: un paquete de las deliciosas Frutigran con chips de chocolate. Todo eran risas hasta que Tomás exclamó: *jestas son las **masitas** más ricas del mundo!*. En ese momento Mariano, furioso, comenzó a mencionar una serie de argumentos de por qué se le debe decir *galletita* en vez de *masita*.

Fue una discusión acalorada. Uno de ellos mencionaba que, debido a su naturaleza industrial, debía llamarse *galletita*, mientras que el otro hacía alusión a la condición de su amigo como persona que vive en la capital para argumentar por qué decirle *masita* es defender las raíces pueblerinas.

Finalmente, y sin arribar a alguna conclusión, comenzaron a disfrutar de las ricas Frutigran. Entonces, se dieron cuenta que ambos, por fuera de esta discusión trivial, tienen algo en común: la Frutigran se come pedacito a pedacito, desde afuera hacia adentro.

Para formar una Frutigran, se coloca un pedacito en el centro con forma de hexágono. Luego, se rodea a este con nuevos pedacitos hexagonales de forma que se toquen con los vértices del inicial. Finalmente, se repite este proceso hasta generar la cantidad de capas necesarias, estando compuesta siempre su primer capa de un solo pedacito.



Frutigran de tres capas.

Los amigos le solicitan que realice un programa que, dada la cantidad de capas de la Frutigran, calcule cuántos pedacitos deberán comer con su forma peculiar para consumirla por completo.

Entrada

La entrada consta de múltiples líneas. La primer línea contiene un entero $1 \leq t \leq 10^5$, que representa la cantidad de casos de prueba. Cada caso consiste en un entero $1 \leq n \leq 10^9$, indicando la cantidad de capas que tiene la Frutigran.

Salida

Por cada caso de prueba, deberá imprimir la cantidad de pedacitos que deberán comer Tomás o Mariano para comer la Frutigran por completo.

Ejemplo

| entrada estándar | salida estándar |
|------------------|-----------------|
| 3 | 1 |
| 1 | 7 |
| 2 | 61 |
| 5 | |

Problema G. Salvación de destreza

Autor: *Fernando Sebastián Lomazzi*
Tiempo límite: 1 segundo

¡Rayos y centellas! Después de meses de jugar una campaña del famoso juego de rol de mesa *Trasgos y Mazmorras*, Rombo activó una trampa y fue carbonizado por una bola de fuego. Si tan sólo hubiera pasado esa tirada de salvación de destreza para esquivar la llamarada que se le aproximaba...

Siempre con la mente en el pasado, Rombo empieza a cuestionarse cuáles eran realmente sus oportunidades para lograr esa tirada de salvación, dándose cuenta que no es nada trivial. Él tenía desventaja al estar sobre un terreno inestable pero, por otra parte, tenía un modificador de salvación de destreza realmente alto al ser un pícaro...

Una tirada básica de salvación de destreza con dificultad d se realiza de la siguiente forma: Se tira un d20 (un dado de veinte caras). Si el resultado es un 1, la tirada falla automáticamente; si es 20, es un éxito inmediato. En cualquier otro caso, se suma el modificador de salvación de destreza m al resultado del dado. Si la suma es al menos la dificultad d el resultado es exitoso, en caso contrario, es fallido.

Existe un caso particular para esto: en caso que la tirada tenga ventaja o desventaja. En estos casos, dos d20 son tirados en vez de uno, pero solo el mayor (para ventaja) o el menor (para desventaja) es considerado para el cálculo final.

Rombo, recordando haber activado múltiples trampas durante la campaña, le solicita que escriba un programa que, dado los datos básicos para una tirada de salvación de destreza (la dificultad, el modificador de destreza y si poseía ventaja o desventaja), calcule las probabilidades de que esa tirada sea exitosa.

Entrada

La entrada consta de múltiples líneas. La primer línea contiene un entero $1 \leq t \leq 2000$, que representa la cantidad de casos de prueba. Cada caso consiste en dos líneas, La primer línea contiene dos enteros $0 \leq d \leq 30$ y $-10 \leq m \leq 10$, la dificultad de la tirada y el valor del modificador de la característica. La segunda línea contiene una palabra indicando si la tirada tiene ventaja o desventaja. Esta palabra es "normal" (en caso que no haya ventaja ni desventaja), "ventaja" o "desventaja".

Salida

Por cada caso de prueba, deberá imprimir la probabilidad de que la tirada de salvación de destreza sea exitosa. La salida siempre tendrá cinco dígitos después del punto decimal.

Ejemplos

| entrada estándar | salida estándar |
|--|--------------------|
| 1 12 4 normal | 0.65000 |
| 2 10 3 ventaja 5 -5 desventaja | 0.91000 0.30250 |

Problema H. Ta te ti

Autor: *Fernando Sebastián Lomazzi*
Tiempo límite: 1 segundo

Alicia es una alegre niña que se encuentra en una extravagante fiesta del té junto a sus curiosos amigos, *el Sombrero Loco* y *el Conejo Blanco*. Entre risas y charlas sin sentido, Alicia propone jugar a un juego de su tierra, el *Ta te ti*. Al escuchar un nombre tan extravagante y, al no tener idea de qué se trataba, sus amigos le pidieron que les enseñe a jugar.

Para dar comienzo al juego, es necesario colocar tantas sillas como participantes alrededor de la mesa redonda del té y cada jugador deberá elegir una para sentarse. Luego Alicia dice una frase, en este caso escogió la típica del juego: “Ta te ti suerte para mí, si no es para mí será para ti, ta te ti”. Finalmente, por cada sílaba de la frase y, partiendo siempre de la misma silla inicial, se va señalando a un jugador en sentido horario. El jugador que esté siendo apuntado cuando se diga la última sílaba de la frase será el ganador.

Una sílaba maravillosa es toda sílaba compuesta de dos letras del alfabeto inglés, mientras que una palabra maravillosa es una concatenación de sílabas maravillosas. Así *ga-to* es una palabra maravillosa, pero *pe-rro* no lo es. Como Alicia es aún una niña, existen palabras que le son difíciles de pronunciar, aún así, como ella es una niña maravillosa, todas las palabras que dice son maravillosas.

Alicia, ni tonta ni perezosa, le solicita su ayuda para que genere un programa que, dada la cantidad de participantes y la frase dicha por ella, le diga qué silla debe escoger para ganar el juego. Se considera la primer silla la seleccionada para comenzar el juego.

Entrada

La entrada consta de dos líneas. La primer línea contiene dos enteros $1 \leq p \leq n$ y $1 \leq n \leq 10^5$, la cantidad de participantes del juego y el tamaño de la frase dicha por Alicia. La segunda línea contiene la frase dicha por Alicia compuesta de letras del alfabeto inglés, espacios, comas y puntos.

Salida

La salida consta de una línea, la posición en sentido horario de la silla que debe escoger Alicia para ganar el juego.

Ejemplos

| entrada estándar | salida estándar |
|---|-----------------|
| 3 63 Ta te ti sute pada mi, si no es pada mi seda pada ti, ta te ti. | 1 |
| 5 43 En la casa de Espinete, todo suma hata sete | 2 |

Problema I. Semejanza

Autor: *Lucio Rivera*
Tiempo límite: 1 segundo

Ezequiel, flamante estudiante de Ingeniería en Sistemas de Información en la UTN, ha enfrentado a lo largo de su vida incontables burlas por parte de sus compañeros debido a la gran cantidad de dibujos animados a los que Ezequiel se asemeja. Entre todas las comparaciones que ha experimentado, una en particular parece atormentarlo por encima de las demás: Ezequiel y el pez *Óscar* de *Pecezuelos* son prácticamente idénticos. Las pocas diferencias que puedan existir entre ellos resultan prácticamente imperceptibles para el ojo humano.

Agotado por el tormento que enfrenta a diario, Ezequiel ha invertido incansables esfuerzos en investigar diferentes algoritmos de comparación de imágenes. Su objetivo es demostrar que no existe una similitud real entre él y Óscar, basándose en una métrica específica al comparar una fotografía suya con una imagen de Óscar.

El algoritmo que Ezequiel ha seleccionado para llevar a cabo su defensa se llama *FACIL* (*Face Analysis and Color Interpolation Layers*). La métrica FACIL se basa en contar la cantidad de columnas compuestas exclusivamente por ceros, luego de sustituir cada elemento por el resto del cociente entre dicho elemento y el valor mínimo de su fila.

Con la esperanza de que Ezequiel deje de sufrir, se le solicita a usted que escriba un programa capaz de calcular la métrica FACIL de una imagen cuadrada.

Entrada

La entrada consta de múltiples líneas. La primer línea contiene un entero $1 \leq n \leq 1000$, el tamaño en píxeles de la imagen cuadrada sometida al algoritmo. Las siguientes n líneas contienen n enteros $1 \leq p_{ij} \leq 1000$, representando el valor del pixel ubicado en la intersección entre la fila i y la columna j .

Salida

La salida consta de una línea, la métrica FACIL de la imagen dada.

Ejemplos

| entrada estándar | salida estándar |
|--|-----------------|
| 4 2 7 6 256 27 3 9 11 5 10 5 125 16 15 4 5 | 2 |
| 3 2 7 5 3 1 9 4 3 4 | 0 |

Problema J. El ritual sobre Innsmouth

Autor: *Fernando Sebastián Lomazzi*
Tiempo límite: 1 segundo

En el siniestro y antiguo puerto marítimo del pueblo de *Innsmouth*, perdido entre brumas y secretos cósmicos, se lleva a cabo un ritual de invocación a un *dios primigenio*. En medio de chillidos desgarradores y cantos lúgubres se encuentran los fieles seguidores frente a un viejo tablero, adornado con extraños símbolos y esculpido en madera centenaria. Encima de este, se encuentran una serie de amuletos con poderes paganos en forma de anillo.

Bajo la tenue luz de unas velas, el conductor del ritual, debe elegir un amuleto y dárselo a un seguidor del culto, luego, otro amuleto será absorbido por el dios primigenio. Esto se repite hasta que, en algún momento, no queden más amuletos sobre el tablero. Cada amuleto posee un poder ancestral y, tanto los cultistas como el dios primigenio, seleccionarán sus amuletos de manera que maximicen su poder ancestral.

Como periodista estudioso en las artes ocultas, te encuentras en una situación extremadamente riesgosa. Tienes el desafío de descubrir el alcance del poder ancestral que el culto podría desatar, ya que conoces detalles sobre los amuletos que serán utilizados en el aterrador ritual. *¡Iä-R'lyeh! ¡Cthulhu fhtagn! ¡Iä! ¡Iä!. Que esta advertencia sea escuchada por quienes se atrevan a buscar lo desconocido, pues el precio de la curiosidad puede ser la perdición.*

Entrada

La entrada consta de dos líneas. La primer línea contiene un entero $1 \leq n \leq 10^4$, la cantidad de amuletos sobre el tablero. La segunda línea contiene n enteros $0 \leq p_i \leq 10^5$, representando el poder ancestral del amuleto i .

Salida

La salida consta de una línea, la cantidad máxima de poder ancestral que puede lograr el culto.

Ejemplos

| entrada estándar | salida estándar |
|-------------------|-----------------|
| 1 777 | 777 |
| 5 10 15 10 5 4 | 29 |

Problema K. Querido Watson

Autor: *Fernando Sebastián Lomazzi*
Tiempo límite: 1 segundo

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 caso de prueba, deberá imprimir la cantidad diferentes de bacterias que existen en el rango $[a, b]$.

Ejemplo

| entrada estándar | salida estándar |
|------------------|-----------------|
| 5 3 | 2 |
| 3 2 3 1 2 | 3 |
| 1 3 | 3 |
| 2 4 | |
| 1 5 | |

Problema L. Linaje

Autor: *Lucio Rivera*
Tiempo límite: 1 segundo

De todas las dinastías influyentes en la historia argentina, es probable que la familia *Maldomado* sea la más destacada y reconocida de todas ellas. Los fundadores de esta familia, distinguidos ciudadanos ilustres de origen europeo, se caracterizaron por su profundo apego a las tradiciones familiares y preservación de su reconocido linaje. Estos valores han perdurado a lo largo de las generaciones futuras de la dinastía.

Nachain Maldomado XVII, un descendiente lejano del fundador *Dieguinho Maldomado IV* y hermano del legendario *Totein Maldomado XVII*, ha advertido que no todas las familias de la dinastía poseen la misma pureza de sangre. Por lo tanto, ha indagado y posteriormente descubierto un método para evaluar la pureza de una familia Maldomado en función de los nombres de sus miembros vivos.

El método propuesto por *Nachain Maldomado XVII* consiste en obtener la longitud del sufijo común más largo entre los nombres de los miembros de la dinastía en consideración sin diferenciar entre mayúsculas y minúsculas, siendo este su nivel de pureza. Cuanto más largo sea ese sufijo, mayor será la pureza de la familia que lleva con orgullo el apellido Maldomado.

Entrada

La entrada consta de múltiples líneas. La primer línea contiene un entero $1 \leq n \leq 1000$, que representa la cantidad de miembros en la dinastía. Las siguientes n líneas contienen un entero $1 \leq |s_i| \leq 1000$, representando el tamaño del nombre del miembro i -ésimo, y una cadena de caracteres s_i , representando el nombre del miembro i -ésimo, compuesta de letras mayúsculas y minúsculas del alfabeto inglés.

Salida

La salida consta de una línea, el nivel de pureza de sangre de la familia Maldomado dada.

Ejemplos

| entrada estándar | salida estándar |
|---|-----------------|
| 2 7 Nachain 6 Totein | 2 |
| 4 5 messi 4 enzo 8 riquelme 7 lomazzi | 0 |

Problema M. Medias

Autor: *Fernando Sebastián Lomazzi*
Tiempo límite: 1 segundo

Elidan Tromba es un millennial amante del frío, cada tarde toma un café negro junto al fuego de su chimenea a leña mientras admira la belleza de la nieve en su ventana. Su amor es tan, tan profundo que hasta se autoproclama un *Team Invierno*. Además, Elidan es un amante de la ropa de hilo, por lo que tiene suéters, medias, gorros y guantes de todo tipo y color.

El señor Tromba se despertó una mañana fresca como cualquier otra en *Siberia*, su ciudad, con una sensación térmica de -20°C . Calentito en su cama y con la luz apagada, Tromba, buscó en su cajón un par de medias del mismo tipo para vestirse. Al no poder ver debido a la falta de luz y, como levantarse de su cama no era una opción, se preguntó ¿Cuántas medias debería sacar del cajón para asegurarme de tener al menos un par de medias del mismo tipo?

Como Elidan es muy perezoso quisiera sacar la cantidad mínima de medias (ya que volver a ponerlas al cajón requeriría un esfuerzo enoorme), y, al poseer una memoria extraordinaria, se acuerda a la perfección la cantidad de medias que posee de diferentes tipos en el cajón.

Ayuda a Elidan a poder ponerse las medias con el mínimo esfuerzo y sin pasar frío.

Entrada

La entrada consta de dos líneas. La primer línea contiene un entero $1 \leq m \leq 10^5$ que representa la cantidad de tipos de medias diferentes en el cajón. La segunda línea contiene m números enteros $1 \leq m_i \leq 10^9$, representando la cantidad de medias del tipo i en el cajón.

Salida

La salida consta de una línea, la cantidad mínima de medias que se deben sacar del cajón para asegurarse tener al menos un par del mismo tipo o “-1” (sin las comillas) en caso que no sea posible realizar lo solicitado.

Ejemplo

| entrada estándar | salida estándar |
|------------------|-----------------|
| 3 2 4 2 | 4 |