Problema - 1/998001 =?

Una relación importante de las matemáticas es 1/998001, que genera un resultado de precisión infinita, lo curioso es que es una serie del 0 al 999, escrito en tres cifras, una vez que termina, inicia de nuevo la serie:

```
1/998001 = 0.000001002003004005006007008009010011012013014015016017018019020021
022023024025026027028029030031032033034035036037038039040041042043044045046047048
049050051052053054055056057058059060061062063064065066067068069070071072073074075
076077078079080081082083084085086087088089090091092093094095096097098099100101102
103104105106107108109110111112113114115116117118119120121122123124125126127128129
130131132133134135136137138139140141142143144145146147148149150151152153154155156
157158159160161162163164165166167168169170171172173174175176177178179180181182183
184185186187188189190191192193194195196197198199200201202203204205206207208209210
211212213214215216217218219220221222223224225.........
```

Cálculus les ha pedido a sus estudiantes que resuelvan el problema sin usar una calculadora por lo que deben hacer el trabajo manualmente y entregarle sus respuestas. Cálculus ya no se preocupa cómo encontrar el resultado, sino más bien en verificar que el resultado sea correcto. Es así que su tarea es cerciorarse de que los términos de la relación se encuentren en las propuestas de resultado(truncadas) que le generan sus estudiantes, en caso de que no sea así su nota dependerá de la cantidad de términos faltantes y la precisión de la propuesta. Por lo que necesita saber qué términos no se encuentran en la respuesta si es incorrecta, caso contrario si la respuesta es correcta debe emitir un informe de "CORRECTO".

Por ejemplo, si uno de sus estudiantes le entrega la siguiente respuesta:

`0.000001002003004005006007008009010011012013014015016017018020021022023`, debera indicar los términos que faltan, en este caso particular falta el `019`.

Tu tarea es identificar si una propuesta de solución a la división propuesta es correcta, en este caso debes emitir "CORRECTO", en caso de que no sea hay que indicar los términos que faltan.

Entrada

La entrada consiste de un número entero n que indica en número de casos de prueba, le siguen n líneas y cada una de ellas es una propuesta de solución s que representa a un número real de la forma 0.*ddddddd*..., la cantidad de cifras de s incluido el . no excede a 20000 caracteres y no es menor que 3.

Salida

La salida por cada caso de prueba se muestra en una línea, la misma que contiene la cadena "CORRECTO" o una lista de términos faltantes separados por un espacio en blanco.

Ejemplos de entrada

```
5
0.000001002003004005006007008009010011012013014015016017018020021022023
0.000001003004005007
0.00000100200300400500600700800901
0.00001
0.0000010
```

Ejemplos de salida

```
019
002 006
CORRECTO
001 002 003 004 005 006 007 008 009
CORRECTO
```

Tarea 1: 30 puntos

Los casos de entrada tendrán cantidad de dígitos después del punto decimal equivalente a un m'ultiplo de 3. Por ejemplo:

```
0.000010
```

Tarea 1: 70 puntos

Los casos de entrada tendrán cualquier cantidad de dígitos. Por ejemplo:

```
0.00000101
```

Problema - Inverso factorial

Se define al factorial de un número N, como el producto de todos los números entre 1 a N: `N! = 1 * 2 * 3 * 4 * ... * N` con $0! = 1$.

Por ejemplo `5! = 1 * 2 * 3 * 4 * 5 = 120`

Lo que se pide en este problema, como el título dice, es calcular el inverso del factorial, en otras palabras encontrar un número tal que aplicando el factorial es igual al número de la entrada.

Por ejemplo: el inverso de 120 es 5, ya que $5! = 120$, el inverso de 6 es 3 ya que $3! = 6$

Entrada

Consta de varios números que tienen entre 1 y 10^6 dígitos, estos son factoriales válidos. La entrada termina cuando no existen más datos. Los números son mayores a 1.

Salida

Para cada número que recibas en la entrada le corresponde una línea en la salida, que es el cálculo del inverso del factorial.

Ejemplos de entrada	**Ejemplos de salida**
120 6	5 3

Tarea 1: 30 puntos

La entrada son números entre 1 y 18 dígitos.

Tarea 2: 70 puntos

La entrada son números entre 1 y 10^6 dígitos.

Sugerencias:

1. Usar lectura rápida, si no sabes que es eso agrega esto a tu código:

`ios::sync_with_stdio(0); cin.tie(0);`

en las primeras líneas del main.

Ejemplo:

```
#include <iostream>
using namespace std;
int main(){
   ios::sync_with_stdio(0);
   cin.tie(0);
}
```

2. No usar *endl* como salto de línea, usar "`\n`" en vez de esto.

3. Recuerda leer hasta fin de archivo son varios números por caso de entrada. Esto es intencionado.

Problema - Jim Botón

No hace mucho Jim y Lucas se fueron para China, si para China, con la esperanza de ser maquinistas (a ellos les gusta conducir trenes) .

China es un país muy grande y los niños dicen que son muy despiertos a temprana edad, cosa que es verdad. Jim y Lucas han encontrado al pequeño Pi, que les ha contado que la princesa ha sido secuestrada por Dragones, si dragones de China. Con esta información Jim y Lucas han buscado al Emperador de China para ofrecer rescatar a la princesa a cambio de un permiso para ser maquinistas en la Gran China. Lamentablemente el Gran Bonzo Pin Pao no está contento, nadie puede ser capaz de rescatar a la princesa salvo el Gran Bonzo.

Pin Pao se ha hecho cargo de verificar que Jim y Lucas son capaces de rescatar a la princesa, y para esto ha convencido al Emperador de entregar a los extranjeros palillos de distintos tamaños, los tamaños posibles son de 0 Li, 1 Li, 2 Li, ..., 9 Li; se les dará hasta siete palillos para los que deben calcular cuantas configuraciones de palillos conforman un número primo. ¡Si se equivocan se les cortará la cabeza!.

Pin Pao es un Bonzo muy tramposo, como verás hay ocasiones que da menos palillos de los que debiera a Jim y Lucas, de manera que fallen las configuraciones que el Emperador espera; en estas ocasiones les entrega pequeñas bolitas que representan a los palillos de longitud 0 Li.

¿Puedes ayudar a Jim y Lucas a resolver este problema rápidamente?

Entrada

La entrada consiste de un valor entero positivo n que representa el número de casos que deben resolverse. En las líneas siguientes se presentan los valores en Li de los siete palillos como una secuencia de cifras de una longitud máxima de 7 y mínima de 1.

Salida

Imprimir, por cada caso de prueba, una línea con el número de configuraciones que representan un número primo.

Por ejemplo, para el caso de recibir dos palillos uno de 3 Li y otro de 1 Li (31), se pueden configurar de las siguientes maneras: 1, 3, 13, 31; por tanto hay 3 posibles números primos de estas configuraciones, y la respuesta es 3.

Nota que el 0 no es significativo cuando está más a la izquierda!.

Ejemplos de entrada	**Ejemplos de salida**
4 5 11 31 031 9999	1 1 3 4 0

Tarea 1: 40 puntos

Para esta tarea se considera soluciones con cinco palillos con valores no repetidos para los palillos. Los valores posibles para los palillos van de 1 LI a 9 Li ($[1, 9]$, no aparece el valor 0)

Tarea 2: 60 puntos

Se considera soluciones para palillos de 0 Li a 9 Li ($[0, 9]$). Se considera soluciones con hasta 7 palillos.

Problema - Agua Sucia

Es el 2020 después de carnavales y hay escasez agua. Una nueva empresa de agua sucia se ha fundado y te han contratado. Esta empresa compra y vende turriles que tienen una cierta cantidad de litros de agua sucia. Como el mercado de venta y compra de agua sucia es muy competitivo debido a la agresiva competencia que existe, quieren saber en todo momento si son o no la mejor empresa de venta y compra de agua sucia. La empresa empieza con 0 turriles y se pondrá a comprar y vender turriles a lo largo del tiempo.

El valor de la empresa en la bolsa de valores es igual a la multiplicación de las cantidades en litros de todos sus turriles, es decir, si tienen los turriles con cantidades de agua sucia en litros iguales a $\{1, 5, 2.5\}$ entonces el valor en la bolsa de la empresa es igual a $1 * 5 * 2.5 = 12.5$.

La empresa sabe que es la mejor en su tipo, si es que no puede existir otra empresa que tenga la misma cantidad total de agua (la suma en litros de todos los turriles) y tenga un mejor valor en la bolsa. Por ejemplo la empresa con los barriles $\{2, 2\}$ es mejor en la bolsa que la empresa con los turriles $\{1, 3\}$.

Tu trabajo es decidir despues de cada transacción si la empresa es la mejor o no.

Entrada

La primera línea de entrada contiene un entero T que representa el número de casos de prueba. Para cada caso de prueba se tiene un valor N en una sola línea que representa la cantidad de transacciones que la empresa realiza. La siguiente línea tiene N números $x_1, x_2, \ldots x_n$, separados por un espacio, que representan una compra o venta en orden cronológico. Si x_i es positivo significa que compraron un turril de x_i litros de agua sucia, al contrario si x_i es negativo, representa que se vendió un turril de x_i litros de agua. Se garantiza que una empresa no venderá un turril de una cantidad de litros de agua que no tenga y que la empresa jamás tendrá cero turriles excepto al inicio.

Salida

Imprimir en una línea una cadena de N caracteres que tenga en la posición i-ésima una letra "S" si es que la empresa es la mejor despues de la i-ésima transacción o una "N" si no es el caso.

Ejemplos de entrada	**Ejemplos de salida**
1 4 2 3 2 -3	SNNS

Tarea 1: 30 puntos

Se cumple: $1 \leq N \leq 100$

Tarea 2: 70 puntos

Se cumple $1 \leq N \leq 100000$

Problema - Alfabeto

Recientemente acaban de descubrir textos escritos en una lengua antigua no conocida hasta el momento. La misma consta de muchos caracteres que son difícles de expresar de forma impresa, por esta razón se las va expresar como números, pero su abecedario está en otro orden.

Se conocen listas de letras que se cree que están ordenadas alfabéticamente según el orden de dicho abecedario, cada lista consta de m letras distintas. Los expertos en el tema nos piden decidir si es posible que exista un abecedario que cumple las condiciones de ordenamiento para todas las listas.

Si tenemos las siguientes listas: $[1, 2, 3]$, $[4, 5, 6]$, $[2, 4]$,$[6, 2]$, no cumple las condiciones para ser abecedario; la razón es que tomando las tres primeras listas, entendemos que 2 está antes que 6 (siendo el orden $1, 2, 3, 4, 5, 6$); si se introduce la última lista $[6, 2]$, no se podrá definir si 2 va antes o después que 6 .

Entrada

La entrada consiste de un número entero c que indica el número de casos.

A continuación para cada caso se tiene un número entero n que indica la cantidad de listas a comprobar, luego le siguen n líneas y cada una de ellas representa una lista como una secuencia de caracteres, el primer valor m en cada línea indica el número de caracteres que tiene la lista, luego le siguen los caracteres separados por espacios.

Salida

La salida por cada caso de prueba se muestra en una línea, la misma que contiene la cadena “No Existe Abecedario” si hay un abecedario tal que todas las listas tengan sus letras ordenadas alfabéticamente, y “Existe Abecedario” si hay un abecedario.

Ejemplos de entrada	**Ejemplos de salida**
2 5 3 1 2 3 3 4 5 6 2 2 4 2 1 5 2 6 2 3 3 1 2 3 3 7 3 11 2 1 7	No Existe Abecedario Existe Abecedario

Tarea 1: 30 puntos

El número de listas: $1 \leq n \leq 10$ y m la cantidad de elementos de una lista $2 \leq m \leq 10$.

Tarea 2: 70 puntos

El número de listas: $1 \leq n \leq 10^5$ y m la cantidad de elementos de una lista $2 \leq m \leq 10^5$.

Problema - Ositos Barrigones Intergalácticos

Los Ositos Barrigones Intergalácticos (OBI) crearon una red comercial construyendo una ruta, bi-direccional, entre todos los pares de planetas de la galaxia. Todos eran felices realizando sus negocios hasta que los Insectos Corrupto-Planetario-Comerciantes (ICPC) llegaron a corromper las rutas comerciales. Se dice que una ruta es corrupta si los ICPC se apropiaron de esta.

Los ICPC sólo pueden recorrer rutas corruptas y los OBI por rutas no corruptas.

Se dice que un subconjunto de planetas es seguro comercialmente si existe una ruta comercial no corrupta entre cada par de estos planetas. Un subconjunto de un solo planeta es siempre seguro comercialmente y se lo debe tomar en cuenta.

Tu trabajo es responder a las preguntas de los ositos en la configuración actual de la red comercial.

Se realizan dos tipos de operaciones en la red comercial:

1. Pregunta. ¿Cuántos distintos subconjuntos de planetas no vacíos son seguros comercialmente en este momento?
2. Actualización. Los ICPC Corrompen una ruta de la red comercial que conecta el planeta U con el planeta V.

Se garantiza que los ICPC corrompen rutas sólo si esta les permite un nuevo camino entre dos planetas, entre los cuales antes no podían moverse. En otras palabras, jamás corremperán la ruta de U hacia V si ya existía un camino de ruta corrupta entre estos dos planetas.

Entrada

La primer línea de entrada contiene un valor entero T, que representa la cantidad de casos de prueba, con $1 \leq T \leq 10$.

A continuación para cada caso de prueba se tiene: una línea con dos valores enteros N y Q que representan el número de planetas y el número de operaciones respectivamente. Le siguen Q operaciones, una por línea. Una operación de pregunta se representa con la letra b en la línea, una operación de actualización se representa con una letra a seguida de dos números U y V separados por espacios. U y V son la numeración de los planetas y juntos representan el camino de U hacia V que a partir de este momento estará corrompido. Con $1 \leq U \leq N$ y $1 \leq V \leq N$.

Salida

Por cada línea de entrada que es una operación de actualización se debe imprimir la cantidad de distintos subconjuntos de planetas no vacíos que son seguros comercialmente en ese momento.

Ejemplos de entrada	Ejemplos de salida
2 3 4 b a 1 2 a 1 3 b 10 10 b a 4 7 a 5 8 a 5 9 a 5 10 b a 2 4 a 2 5 a 1 3 b	7 4 1023 431 257

Tarea 1: 10 puntos

El número de planetas y operaciones cumplen las siguientes restricciones: $1 \leq N \leq 10$ y $1 \leq Q \leq 15$.

Tarea 2: 20 puntos

El número de planetas y operaciones cumplen las siguientes restricciones: $1 \leq N \leq 100$ y $1 \leq Q \leq 15$.

Tarea 3: 70 puntos

El número de planetas y operaciones cumplen las siguientes restricciones: $1 \leq N \leq 1000$ y $1 \leq Q \leq 2000$.