



UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
FACULTAD DE INGENIERIA ELECTRICA, ELECTRONICA, INFORMATICA Y MECANICA
ESCUELA PROFESIONAL DE INGENIERIA INFORMATICA Y DE SISTEMAS

Concurso de Programación
VIII CusContest

PROBLEM SET

Este conjunto de problemas contiene 9 problemas,
páginas numeradas de 1 a 10.

Organizado por:
ACM Chapter Cusco
EPIIS-UNSAAC
DAI-UNSAAC
CCFF de EPIIS
Curso de Actividades y Producción de Servicios

Diciembre 16, 2015
CUSCO - PERÚ

INFORMACIÓN GENERAL

Salvo se indique lo contrario, lo siguiente vale para todos los problemas.

Entrada

- La entrada se debe leer de la entrada estándar (*standard input*).
- Cuando una línea de datos contiene varios valores, estos se separan utilizando exactamente un espacio entre ellos. Ningún otro espacio aparece en la entrada.
- No hay letras con tildes, acentos, diéresis, ni otros signos ortográficos (ñ, Ã, ç, ú, æ, etcétera).

Salida

- La salida se debe escribir en la salida estándar (*standard output*).
- El resultado del caso de prueba debe de aparecer en la salida utilizando una cantidad de líneas que depende del problema. No debe de haber otros datos en la salida.
- Cuando una línea de resultados contiene varios valores, estos se deben de separar utilizando exactamente un espacio entre ellos. Ningún otro espacio debe de aparecer en la salida.
- No debe de haber letras con tildes, acentos, diéresis, ni otros signos ortográficos (ñ, Ã, ç, ú, æ, etcétera).

Tiempo Límite

- El tiempo límite informado corresponde al tiempo total permitido para la ejecución de una solución sobre todo el conjunto de casos de prueba de un determinado problema.

Equipo de desarrollo:

Alexander Pinares, Federal University of Rio Grande do Sul
Berthin Torres, University of Campinas
Edu Sánchez, University of São Paulo
Grover Castro, University of São Paulo
Waldir Farfán, University of São Paulo

Problem A. Piedras de Saqsayhuaman!

Input file: Standard input
Output file: Standard output
Time limit: 1 second
Memory limit: 64 megabytes

Killa y Chaska fueron a visitar la fortaleza de Saqsayhuaman. Ellas, muy curiosas, disponían de mucho tiempo y decidieron medir la altura de cada una de las piedras del centro arqueológico en mención.

Regresando de su largo paseo, se encontraron con su amigo Ramón y le mostraron una lista que contenía las medidas de todas las piedras. Ramón, un joven que gusta mucho de las matemáticas, inmediatamente pensó si sería posible obtener de dicha lista un conjunto libre de K -múltiplos. Un conjunto libre de K -múltiplos está compuesto por números distintos tal que no existen dos elementos X y Y ($X < Y$) que puedan cumplir con la siguiente ecuación: $X * K = Y$

Ramón requiere tu ayuda para resolver su problema, pues él quiere determinar el mayor conjunto libre de K -múltiplos dada la lista de alturas.

Input

El problema contiene varios casos de prueba. La primera línea es un entero T ($1 \leq T \leq 10^2$) que denota el número de casos de prueba. Cada caso está compuesto por dos líneas, en la primera de ellas se encuentran los números enteros N y K , siendo N el número de piedras medidas ($1 \leq N \leq 10^5$, $1 \leq K \leq 10^9$). Luego, en la segunda línea del caso de prueba se encuentran N enteros distintos a_1, a_2, \dots, a_N que representan las alturas de las piedras ($1 \leq a_i \leq 10^9$).

Output

Para cada caso de prueba, el programa deberá imprimir la longitud del mayor conjunto libre de K -múltiplos que se pueda obtener de la lista de alturas.

Example

Standard input	Standard output
2	1
2 3	3
1 3	
6 2	
2 3 6 5 4 10	

Problem B. Gotta catch'em all!

Input file: Standard input
Output file: Standard output
Time limit: 1 second
Memory limit: 64 megabytes

En una pequeña ciudad llamada Pueblo Paleta vivía Red junto a sus amigos Green y Blue. Todos eran amantes de los pokémon y siempre luchaban por ser quienes capturen y entrenen a los mejores pokémons. Nuestro amigo Red decía que no le importaba mucho capturar a los mejores pokémons porque al final con un buen entrenamiento cualquier pokémon podría ser capaz de salir victorioso en un combate. Por ello, sólo gustaba capturar a los pokémon cuyo nombre cumpla una única regla la cual era: si el caracter con mayor número de repeticiones era uno de los caracteres contenidos en el nombre de Red entonces lo capturaba, caso contrario simplemente lo dejaba escapar.

Dado que, considerando las condiciones expuestas por Red, la probabilidad de encontrar un pokémon con dichas características es muy baja y más aún con el poco tiempo que tiene para decidir si capturarlo o no, Red necesita de alguna forma adicionar una aplicación a su Pokédex para que al momento de consultar sus datos aparezca un mensaje para saber si cumple o no sus condiciones.

Como todo buen(a) pokemaniático(a), tu deber es ayudar a Red diseñando el aplicativo para la Pokédex. Considerar que el nombre de un pokémon es una cadena de caracteres conformada por símbolos ASCII, únicamente letras del alfabeto inglés, pueden ser minúsculas y/o mayúsculas. Siempre considerar como diferentes mayúsculas y minúsculas, es decir 'A' es diferente de 'a'.

En ciertos casos, pueda que haya más de un caracter con el máximo número de repeticiones, por ello, si ocurre un empate se verificará la condición de Red con cada uno de esos caracteres. Al verificar no es necesario distinguir mayúsculas de minúsculas. Si por ejemplo, se tuviese como dato de entrada **Charmander**, los caracteres con mayor número de repeticiones son 'a' y 'r', y como Red contiene una 'R', la pokédex debería de indicar que si es posible capturar a **Charmander**.

Input

El problema contiene varios casos de prueba, y cada caso de prueba está conformado por una cadena de caracteres $S = c_1, c_2, c_3, \dots, c_N$ de longitud N ($0 < N \leq 10^2$). Dado que Arceus es el dios pokémon, al encontrar este caso de prueba (ya sea todas minúsculas, mayúsculas, o una mixtura de ambas) inmediatamente la pokédex se apagará y ya no se procesarán más datos de entrada.

Output

Para cada caso imprimir en una línea la entrada y un mensaje de respuesta (YES o NO) separados por un espacio en blanco. YES indicará que Red puede y debe de capturar al pokémon, y NO que debe dejarlo escapar.

Example

Standard input	Standard output
Pikachu	Pikachu NO
CharIZARD	CharIZARD YES
Bulbasur	Bulbasur NO
Charmander	Charmander YES
ARCEUS	
Moltres	

Problem C. Codificación

Input file: Standard input
Output file: Standard output
Time limit: 1 second
Memory limit: 64 megabytes

Xavier, un experto estudioso en el tema de convulsiones (contracciones involuntarias de los músculos), recién encontró unos antiguos manuscritos de una civilización muy poco conocida. Los 'Wantans', que era así como se llamaba la civilización, se caracterizaba por numerosos estudios en el campo de la medicina y astrología.

Cuando Xavier estaba a punto de comenzar con su investigación, se dio con la sorpresa de que los manuscritos tenían algún tipo muy extraño de codificación. Debido a ello, inmediatamente se puso en contacto con otros investigadores de la Universidad de Marle Baar quienes habían encontrado una posible solución para poder resolver el sistema de codificación usado por los Wantans.

Básicamente, el código es un conjunto de símbolos que a simple vista no tenían sentido alguno, pero al ordenar los símbolos siguiendo una secuencia se podía construir una oración. Los investigadores encontraron el patrón y el conjunto de símbolos, pero no son tan hábiles como para manualmente realizar la tarea de decodificación. Por ello, solicitaron tu ayuda para diseñar un algoritmo que reciba un conjunto de símbolos (que puede ser interpretado como una matrix rectangular $W \times H$ de caracteres) y retorne la oración decodificada siguiendo el patrón descrito en la Figura 1a. Por ejemplo, la Figura 1b muestra uno de los códigos que Xavier encontró, y siguiendo el patrón antes mencionado se puede obtener el siguiente mensaje: `Concurso#de#Programación#VIII#CusContest#ACM#Chapter#CUSCO#2015#`.

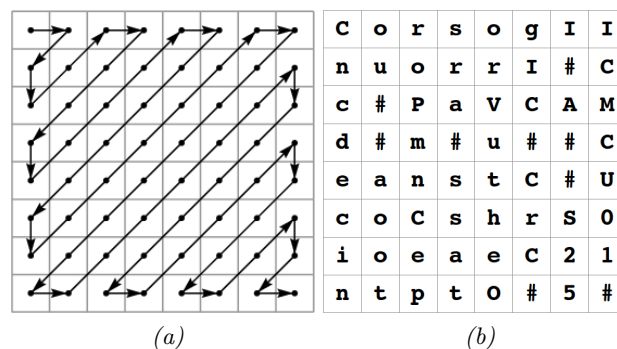


Figura. 1: (a) El patrón encontrado por los científicos. (b) Uno de los muchos códigos antiguos encontrados por Xavier.

Cabe señalar que los Wantans no conocían el uso del espacio en blanco, y en su lugar empleaban el carácter *hashtag* #.

Input

El problema está conformado por varios casos de prueba. La primera línea contiene un número T ($1 \leq T \leq 10^2$) indicando el número de casos de prueba. Cada caso comienza con una línea con dos números enteros H W ($1 < H, W \leq 10^2$) que indican la altura y el anchura de la matriz de caracteres que contiene el código encontrado por Xavier. Las siguientes H líneas contienen W caracteres. Los caracteres pueden ser letras del alfabeto, dígitos, como también *hashtags*.

Output

Para cada caso de prueba, imprimir en una línea el mensaje decodificado.

Example

Standard input	Standard output
2 2 2 Ab cd 7 5 Esoro t#an# ac#tc iniuh oemot n##ni esedo	Abcd Esta#oracion#no#tiene#mucho#sentido

Problem D. La instrumentalización humana

Input file: Standard input
Output file: Standard output
Time limit: 1 second
Memory limit: 64 megabytes

Seele, una misteriosa organización de poder global conformada por varios países como USA, China y Alemania; posee unos escritos antiguos conocido como *Los rollos del mar muerto* pertenecientes a los primeros seres vivos de la Tierra, Lilith y Adam. Estos rollos describen muchos acontecimientos que ya pasaron y van a pasar.

Gracias a estos manuscritos se llegó a saber sobre la instrumentalización humana, y Seele tiene como objetivo completar dicho proyecto para llevar a la humanidad a un siguiente paso en la evolución donde no hay dolor ni maldad, todos seremos felices. Pero, para llegar a completar esta tarea es necesario descubrir los secretos de los manuscritos de mar muerto de Lilith.

Tú eres un(a) trabajador(a) de esta misteriosa organización y estás a cargo de la doctora Yui Ikari, ella es una bio-ingeniera que encargada de la creación de bio-androides conocidos como EVAS. Tras años de investigación, la doctora logró entender el mensaje de los manuscritos, el cual dice: *"Para lograr ser uno y vivir eternamente debes dejar tu yo y ser parte de mí..."*. Junto al mensaje se encontró una ecuación (N^M) y un número K . Yui cree que la forma de unir un EVA con un ser humano está dado por los primeros K dígitos de N^M , pero su habilidad para programar no es tan buena como la tuya y por ello te pide que la ayudes en su investigación.



Figura. 2: Dra. Yui Ikari

Input

La entrada está conformada por varios casos de prueba. La primera línea contiene un entero positivo T ($T < 20$) indicando el número de casos de prueba. Las siguientes T líneas contienen los tres enteros positivos N, M, K ($1 \leq N, M \leq 10^9, 1 \leq K \leq 10$) encontrados junto al *Rollo del mar muerto*.

Output

Para cada caso de prueba imprimir en una línea los primeros K dígitos de N^M . Está garantizado que N^M tiene más o igual cantidad de dígitos que K .

Example

Standard input	Standard output
2	781
5 7 3	794968
12 24 6	

Problem E. Buscando la vaca!

Input file: Standard input
Output file: Standard output
Time limit: 1 second
Memory limit: 64 megabytes
Original idea: Brian Dean

Bessie la vaca ha escapado y ahora está escondida en una colina cubierta de hierba. El granjero John, en un intento de recuperar a Bessie, decidió arrastrarse por la hierba de manos y rodillas para que pueda acercarse sin ser detectado. Desafortunadamente, él tiene problemas para encontrar a Bessie desde esa posición.

La hierba delante del granjero parece una cadena de N paréntesis, por ejemplo `)(((())())` representa un poco de hierba.

El granjero John sabe que las patas traseras de Bessie se parecen a un par de paréntesis de apertura adyacente `((`, y que sus patas delanteras se parecen a un par de paréntesis de cierre adyacente `)`'. Por lo tanto, la ubicación de Bessie puede ser descrita por un par de índices $x < y$ ($0 < x$) tal que `((` se encuentra en la posición x , y `)`' se encuentra en la posición y .

Tu deber es ayudar a calcular el número de diferentes posibles lugares en que Bessie podría estar.

Input

El problema recibirá multiples archivos de entrada, cada archivo conteniendo un caso de prueba. La primera línea del caso de prueba contiene una cadena de caracteres de tamaño N ($1 \leq N \leq 5 \times 10^4$). Está garantizado que los únicos caracteres de la cadena serán `(` y/o `)`.

Output

Para cada caso de prueba, imprimir en una línea el número posibles de posiciones donde Bessie podría estar. Esto es, el número distinto de pares de índices $x < y$, tal que existe un patron `((` en la posición x y un patrón `)`' en la posición y .

Example

Standard input	Standard output
<code>)(((())())</code>	4

Para este ejemplo, las posiciones en las que Bessie puede estar son: $(2, 7)$, $(3, 7)$, $(2, 9)$ y $(3, 9)$. Por lo tanto la respuesta es 4.

Problem F. Doblando una cuerda

Input file: Standard input
Output file: Standard output
Time limit: 1 second
Memory limit: 64 megabytes
Original idea: Brian Dean

El granjero Juan tiene una cuerda larga de longitud L que usa para varias tareas en la granja. La cuerda tiene N nudos en varias ubicaciones distintas, incluyendo un nudo en cada uno de sus dos extremos.

Juan se da cuenta que hay ciertas ubicaciones en las cuales es posible puede doblar la cuerda sobre si misma de tal manera que los nudos opuestos se alinean exáctamente uno con otro.

Por favor, ayuda al granjero Juan a contar el número de puntos de doblado (ver Figura 3) que tienen esta propiedad.

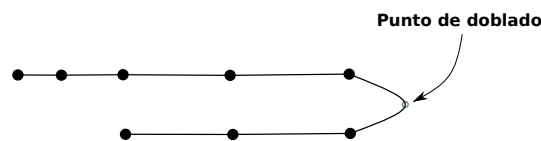


Figura. 3: Punto de doblado que permite a los nudos de la cuerda alinearse unos con otros.

Se permite doblar exáctamente en un nudo, excepto en los puntos extremos. Todos los nudos en el lado más largo no son un problema (eso es, se necesita únicamente que los nudos se alineen donde los dos pedazos de la cuerda coinciden al hacer el doblado). Afortunadamente, Juan toma en cuenta hacer un sólo doblado cada vez y él nunca hace varios doblados.

Input

El problema contiene varios casos de pruebas separados en diferentes archivos. Cada archivo contiene un caso de prueba. La primera línea de cada caso contiene dos enteros N y L separados por un espacio en blanco ($2 \leq N \leq 10^2, 1 \leq L \leq 10^4$). Las siguientes N líneas contienen un entero en el rango $0 \dots L$ especificando la ubicación de un sólo nudo. Dos de estas líneas contendrán siempre 0 y L .

Output

Para cada caso, imprimir el número de posiciones válidas de doblaje.

Example

Standard input	Standard output
5 10 0 10 6 2 4	4

La cuerda tiene longitud $L = 10$, y hay 5 nudos en las posiciones 0, 2, 4, 6 y 10. Por lo tanto, las posiciones válidas de doblaje son 1, 2, 3 y 8.

Problem G. Kywhis

Input file: Standard input
Output file: Standard output
Time limit: 1 second
Memory limit: 64 megabytes

Xavier, un gran matemático, festejó hace poco su cumpleaños tomando una gran cantidad de “kywhis”, su refresco favorito. Después de la fiesta, él y sus amigos decidieron construir M pilas con las latas puestas una encima de otra. Luego, Xavier se propuso a comprobar su puntería derribando las pilas de latas, pero como era una tarea muy fácil (tiene una puntería excepcional a causa de años de experiencia) decidió poner la restricción de que únicamente sea posible derribar las latas que están en la cima de su respectiva pila. Como todo gran matemático, se puso a pensar de cuántas formas podría derribar todas las latas tomando en cuenta la restricción añadida.

Lamentablemente, Xavier se siente muy mal debido a que bebió mucho “kywhis”, y quiere que lo ayudes.

Input

El problema contiene varios casos de prueba. La primera línea tendrá un entero T ($0 < T \leq 10^3$) que indica el número de casos, para cada caso un número entero M ($1 \leq M \leq 10^3$) indicará el número de pilas de latas que Xavier armó junto a sus amigos, seguido por M números a_1, a_2, \dots, a_M ($1 \leq a_i \leq 10^3$), tal que a_i representa el número de latas apiladas en la i -ésima pila.

Output

Para cada caso de prueba imprimir una línea con el número de formas en las que Xavier puede derribar todas las latas, respetando su restricción por supuesto.

Dado que la respuesta puede ser muy grande, deberás imprimir el resultado modulo $10^9 + 7$.

Example

Standard input	Standard output
2	6
2	560
2 2	
3	
3 2 3	

Problem H. Cerca del tercer impacto

Input file: Standard input
Output file: Standard output
Time limit: 1 second
Memory limit: 64 megabytes

Desde el inicio de los tiempos existen los ancestrales. Ellos tienen la misión de crear vida en los planetas, pero la única forma de crear vida es hacer una copia de ellos. El problema es que no quieren crear algo tan poderoso que se les asemeje ya que no podrían controlarlos, por lo que deciden separar su perfecto ser en 2 tipos de semillas.

Una de las semillas es la luna blanca que contiene a Adam, el fruto de la vida eterna. Los descendientes de Adam son llamados ángeles los cuales puede tomar cualquier forma y gozan de la vida eterna. El otro tipo de semilla es la luna negra que contiene a Lilith, el fruto del conocimiento; la cual tiene la capacidad de crear vida como animales y humanos quienes gozan del conocimiento pero con un tiempo corto de vida. Si alguien tuviera las dos semillas podría llegar a tener la capacidad de los ancestrales.

A raíz de esto, sólo una semilla es enviada a cada planeta, y en un supuesto caso en el que se llegasen a encontrar las dos semillas, se abriría un agujero de gusano conocido como la puerta Goaf que destruiría todo el planeta.

Lamentablemente a la Tierra llegaron las dos semillas, pero una se desactivó en la caída. Cada semilla venía junto con un pergamino conocido como *Los rollos de Mar Muerto*. Aunque estos rollos se encuentran encriptados, tienen la capacidad de predecir lo que sucederá en el futuro.

Tras años de investigación uno de los rollos pertenecientes a Adam fue develado, y decía: "*La llegada de los descendientes de Adam es inminente y con ello ocurrirá la destrucción...*". Junto a ese mensaje se encontró una ecuación (N^M) y un número K . Los científicos creen que el tiempo para la llegada de los ángeles está dado por los últimos K dígitos de N^M .

Tú eres un(a) trabajador(a) de NERV, una organización encargada de proteger la Tierra. Tú misión es determinar el tiempo de la llegada de los ángeles para que los trabajadores de NERV puedan prepararse y evitar el tercer impacto.

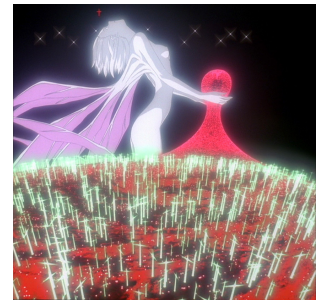


Figura. 4: Visión del tercer impacto: Lilith absorbiendo las almas de toda la humanidad.

Input

La entrada está conformada por varios casos de prueba. La primera línea contiene un entero positivo T ($T < 20$) indicando el número de casos de prueba. Las siguientes T líneas contienen los tres enteros positivos N, M, K ($1 \leq N, M \leq 10^6, 1 \leq K < 10$) encontrados junto al *Rollo del mar muerto*.

Output

Para cada caso de prueba imprimir los últimos K dígitos de N^M . Está garantizado que N^M tiene más o igual cantidad de dígitos que K .

Example

Standard input	Standard output
2	125
5 7 3	441536
12 24 6	

Problem I. Fardos de Heno

Input file: Standard input
Output file: Standard output
Time limit: 1 second
Memory limit: 64 megabytes
Original idea: Brian Dean

¡Las vacas están haciendo otra vez de las suyas! El granjero Juan ha organizado N pilas de fardos de heno, todas de la misma altura. Sin embargo, cuando él no está mirando, las vacas mueven algunos de los fardos entre las pilas, por lo tanto sus alturas ya no son necesariamente iguales.

Dadas las nuevas alturas de todas las pilas, ayuda por favor al granjero Juan a determinar el mínimo número de fardos de heno que él necesita mover con el propósito de restaurar todas las pilas a sus alturas originales (todas iguales).

Input

El problema contiene varios casos de prueba separados en diferentes archivos. Cada archivo contiene un único caso. La primera línea del caso de prueba contiene N ($1 \leq N \leq 10^4$) indicando el número de pilas. Después, N líneas son presentadas, cada una con un entero f_i ($1 \leq f_i \leq 10^4, 1 \leq i \leq N$) representando el número de fardos de heno en una sola pila.

Output

Para cada caso, imprimir un entero dando el número mínimo de fardos que necesitan ser movidos para que todas las pilas vuelvan a tener la misma altura.

Example

Standard input	Standard output
4 2 10 7 1	7

Hay 4 pilas, de alturas 2, 10, 7 y 1.

Moviendo 7 fardos de heno (3 de la pila 2 a la pila 1, 2 de la pila 2 a la pila 4, 2 de la pila 3 a la pila 4), podemos hacer que todas las pilas tengan altura 5.