



UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA,
INFORMÁTICA Y MECÁNICA
ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS

**Concurso de Programación
INTERCON XXIV - 2017**

PROBLEM SET

Este conjunto de problemas contiene 8 problemas,
páginas numeradas de 1 a 8.

Organizado por:

**INTERCON 2017 - CUSCO
ACM CHAPTER CUSCO**

Agosto 17, 2017

CUSCO - PERÚ

INFORMACIÓN GENERAL

A menos que 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 la cantidad de líneas indicada para cada 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
Grover Castro - University of Sao Paulo
Jhosimar Arias - University of Campinas
Kent Yucra - University of Campinas
Kleiber Ttito - Federal University of Rio Grande do Sul
Rodolfo Quispe - University of Campinas
Waldir Farfan - University of Sao Paulo

Problem A. La cerca de Javier

Input file: Standard input
Output file: Standard output
Time limit: 3 segundos
Memory limit: 64 megabytes
Original idea: Berthin Torres

Érase una vez un granjero llamado Javier que solía vivir en los alrededores de Cusco. Él es una buena persona y lleva una vida simple junto a sus animales. Sin embargo, algunas de sus vacas eran rebeldes y solían romper las cercas de Javier – supongamos que las cercas estan construidas como una secuencia de tiras de madera que yacen sobre el suelo, y por motivo de simplicidad, las cercas serán representadas por una secuencia de 0s y 1s, donde 0 indica una tira de madera rota y 1 indica una tira de madera en buen estado.

Debido a este problema, Javier estaba a falta de dinero porque reemplazar las cercas era muy costoso. En vez de reemplazar todas las cercas, él pensó que esta vez podía comprar solo una tira de madera. Ahora el problema es dónde colocarla.

Javier no es conocido por sus habilidades analíticas, entonces solicitó tu ayuda. Lo que Javier necesita básicamente es poner la nueva tira que compró y reemplazar una de las que rompieron sus locas vacas, pero él quiere saber donde reemplazarla y cuál sería la secuencia más larga de la cerca en buen estado.

Input

La primera línea de la entrada contiene un número $T(1 \leq T \leq 100)$ indicando el número de casos de prueba. La primera línea de cada caso de prueba contine un número $N(1 \leq N \leq 1000)$ el cual representa la longitud de la cerca de Javier, y la segunda línea contine N números binarios(separados por espacios) representando la cerca. Supón que hay al menos una tira rota que Javier siempre va a reemplazar usando la tira que compró.

Output

Para cada caso de prueba, imprime una línea mostrando la posición donde Javier debe reemplazar la tira de madera rota y también el tamaño de la secuencia más larga de la cerca en buen estado, después de reemplazarlas por supuesto. La posición debe comenzar en el índice 0. En caso de que pueda poner la tira de madera en más de un lugar, indica el menor índice.

Example

Standard input	Standard output
2	5 2
5	9 8
1 1 0 1 1	
13	
1 1 1 0 1 1 1 1 0 1 1 1 1	

Explicación

Para el primer caso, hay 5 tiras de madera, la 2^{da} posición esta rota. Si Javier la reemplaza, la cerca entera estará bien, tal que la longitud total de la cerca en buen estado es 5.

Para el segundo caso. Javier podría reemplazar la pieza rota en la 3^{ta} posición consiguiendo una cerca con una longitud máxima de la cerca en de 8, sin embargo, si reemplaza la 8^{va}, la longitud máxima de la cerca en buen estado será 9

Problem B. Chusky y el hambre

Input file: Standard input
Output file: Standard output
Time limit: 3 segundos
Memory limit: 64 megabytes
Original idea: Rodolfo Quispe

Chusky vive en Dougland y como todos en la ciudad, él ama el plato tradicional: Aguadito. Hay una única Pollería (un muy buen lugar donde comer aguadito) en Dougland, por lo tanto, este comedor es realmente selectivo con sus comensales. Para comer en la Pollería, Chusky debe resolver la siguiente tarea: Dadas D cajas, cada una tiene todos los números enteros entre L_i y R_i ($1 \leq i \leq D$) incluyéndolos. Indicar el K -ésimo único menor elemento. Debido a que hay una gran cola detrás de Chusky, él tiene pocos segundos para dar la respuesta, de otra manera, él no podrá comer en la Pollería, entonces tendrá que comer las papas hervidas que tenía en casa (no le gustan mucho). Ayuda a Chusky a comer un Aguadito.

Input

La primera línea de entrada contiene un número T ($1 \leq T \leq 100$) indicando el número de casos de prueba. Cada caso de prueba sigue el siguiente formato:

La primera línea de cada caso de prueba contiene un número D ($1 \leq D \leq 100$) representando el número de cajas que Chusky recibe, la segunda línea contiene un número K ($1 \leq K \leq 10^5$) representando el K -ésimo número que Chusky debe adivinar, luego siguen D líneas, cada una de estas tiene 2 líneas L_i y R_i ($1 \leq L_i \leq R_i \leq 10^5$) indicando que la caja i tiene todos los enteros entre L_i y R_i ($1 \leq D \leq 100$) incluyéndolos.

Nota: Los números podrán no ser soportados por una variable de tipo entero, se sugiere usar un entero de 64-bits (C++: long long, JAVA: long)

Output

Para cada caso de prueba, imprime el K -ésimo menor número de entre todas las cajas. Imprime -1 si ningún K -ésimo menor elemento existe.

Example

Standard input	Standard output
2	4
3	-1
2	
3 6	
3 4	
11 13	
1	
7	
1 5	

Explicación

Para el primer caso de prueba, la primera caja tiene los números 3,4,5,6, la segunda tiene a 3,4 y la última a 11,13, entonces todos los números existentes son 3,4,5,6,11,13 y el segundo menor número es 4. En el segundo caso de prueba la caja tiene los números 1,2,3,4,5, así no hay el 7^{mo} menor elemento.

Problem C. R2-D2 y cambios de signo

Input file: Standard input
Output file: Standard output
Time limit: 3 second
Memory limit: 64 megabytes
Original idea: Kleiber Titto

R2-D2 es un robot construido en la UNSAAC por el equipo del prof. Juan Cruz. R2-D2 trabaja en el área de ventas de la corporación E-Cormmerce. Ahora él quiere preparar un reporte del año pasado.

R2-D2 tiene una lista de los ingresos de la compañía. La lista es una secuencia de N enteros. El ingreso total de la compañía es la suma de todos los enteros de la secuencia. R2-D2 decidió efectuar exactamente K cambios de signo a diferentes números en la secuencia. El puede cambiar el signo de un número más de una vez (dos veces, tres veces, depende de el pero está limitado por K).

La operación de cambiar el signo a un número es la de multiplicar este número por -1 .

Ayuda a R2-D2 a efectuar los cambios de tal forma que el ingreso total de la compañía (la suma de los números en la secuencia resultante) sea máximo. Recuerda que R2-D2 debe realizar exactamente K cambios.

Input

La primera línea de la entrada contiene un número T ($1 \leq T \leq 50$) indicando el número de casos de prueba.

La primera línea de cada caso de prueba contiene dos enteros N y K ($1 \leq N, K, \leq 10^5$), indicando cuántos números hay en la secuencia y cuántos cambios debe realizar respectivamente.

La segunda línea contiene una secuencia no decreciente, esta consiste en N enteros a_i ($|a_i| \leq 10^4$).

Los números en las líneas están separados por espacios simples. Nota que la secuencia dada está en un orden no decreciente.

Output

En una sola línea muestra la respuesta al problema. El máximo ingreso que puede obtener después de hacer exactamente K cambios.

Example

Standard input	Standard output
2	3
3 2	1
-1 -1 1	
3 1	
-1 -1 1	

En el primer ejemplo podemos obtener la secuencia $[1, 1, 1]$, entonces el ingreso total es 3.

En el segundo ejemplo, la estrategia óptima es obtener la secuencia $[-1, 1, 1]$, entonces el ingreso total es 1.

Problem D. Torres

Input file: Standard input
Output file: Standard output
Time limit: 3 segundos
Memory limit: Grover Castro

Dahem es un muchacho al cuál le gusta coleccionar tableros de ajedrez de diferentes tamaños y estilos. A él también le gusta crear nuevos puzzles para jugar. Cierta día él estaba tratando de contar todas la formas de poner n torres sin que estas se ataquen unas con otras, considera que los ataques de una torre sólo son en dirección vertical u horizontal como en el ajedrez en un tablero de $N \times N$. Repentinamente una idea para un nuevo puzzle paso por su mente: ¿Que podría pasar si las torres tendrían diferentes colores?. ¿De cuántas maneras pueden colocarse las torres en el tablero de ajedrez? En otras palabras, hay N torres de K colores con c_1 torres del primer color, c_2 torres del segundo color, ..., y c_K torres del K -ésimo color. Él quiere obtener el número de formas de colocar estas torres en un tablero de $N \times N$, tal que ninguna torre ataque a otra del mismo color. Sin embargo, Dahem está enamorado y no puede pensar apropiadamente. ¿Podrías ayudarlo con este trabajo?

Input

La primera línea de la entrada contine un número T ($1 \leq T \leq 10^5$) indicando el número de casos de pruebas. La primera línea de cada caso de prueba contiene 2 enteros N y K ($1 \leq K \leq N \leq 12$), describiendo el tamaño del tablero y el número de colores respectivamente. La segunda línea contiene una secuencia de K enteros c_i ($c_i \leq N$) los cuales representan el número de torres de cada color. Los números en cada línea estan separadas por espacios simples.

Output

Imprime la respuesta del problema en una sola línea. El número de maneras de colocar las torres en un tablero de $N \times N$, tal que ninguna torre pueda atacar a otra del mismo color.

Example

Standard input	Standard output
2 1 1 1 8 3 1 3 4	1 11289600

Problem E. LPSS

Input file: Standard input
Output file: Standard output
Time limit: 3 second
Memory limit: 64 megabytes
Original idea: Genady Korotkevich

El nombre verdadero de este problema, "Longest Palindrome String Search" es muy largo como para entrar en la cabecera de la hoja.

Se te dará una cadena s que consiste solamente de letras minúsculas del ingles. Tienes que hallar la subsecuencia palindrómica lexicográficamente más larga.

Llamaremos a una cadena no vacía $s[p_1p_2 \cdots p_k] = s_{p_1}s_{p_2} \cdots s_{p_k}$ ($1 \leq p_1 < p_2 < \cdots < p_k \leq |s|$) una subsecuencia de la cadena $s = s_1s_2 \cdots s_{|s|}$, donde $|s|$ es el tamaño de la cadena s . Por ejemplo, "abcb", "b" y "abacaba" son subsecuencias de la cadena "abacaba". La cadena $x = x_1x_2 \cdots x_{|x|}$ es lexicográficamente mas larga que la cadena $y = y_1y_2 \cdots y_{|y|}$ si se da que $|x| > |y|$ y $x_1 = y_1, x_2 = y_2, \cdots, x_{|y|} = y_{|y|}$, o que exista un número r ($r < |x|, r < |y|$) tal que $x_1 = y_1, x_2 = y_2, \cdots, x_r = y_r$ y $x_{r+1} > y_{r+1}$. Los caracteres de las cadenas son comparados por su respectivo valor en código ASCII. Por ejemplo, la cadena "ranger" es lexicográficamente mas larga que la cadena "racecar" y la cadena "poster" es lexicográficamente mas larga "post".

La cadena $s = s_1s_2 \cdots s_{|s|}$ es un palíndromo si es igual a su inverso $\text{rev}(s) = s_{|s|}s_{|s|-1} \cdots s_1$. En otras palabras, una cadena es un palíndromo si se puede leer de la misma forma de izquierda a derecha y de derecha a izquierda. Por ejemplo, cadenas palindrómicas "racecar", "refer" and "z".

Input

La primera línea contiene un entero T ($T \leq 10$) indicando el número de casos de prueba. Luego siguen T líneas.

Cada linea contiene una cadena no vacia s que consiste únicamente de letras minúsculas del ingles. La longitud de s no excede los 10 caracteres.

Output

Para cada caso de prueba, imprimir la subsecuencia palindrómicamente mas larga de la cadena s .

Example

Standard input	Standard output
4	rr
radar	wwwww
bowwowwow	s
codeforces	ssss
mississipp	

De todas las subsecuencias de la cadena "radar" las siguientes son palíndromos: "a", "d", "r", "aa", "rr", "ada", "rar", "rdr", "raar" and "radar". La mas larga lexicográficamente es "rr".

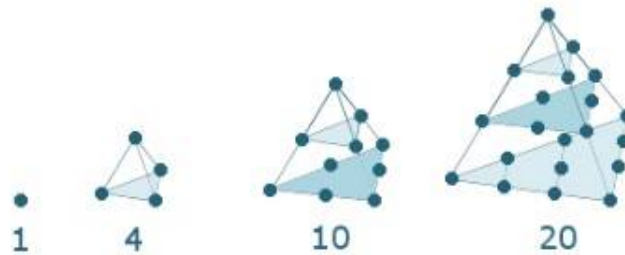
Problem F. Jugando con la conjetura de Pollock Pollock

Input file: Standard input
Output file: Standard output
Time limit: 3 second
Memory limit: 64 megabytes
Original idea: Alexander Pinares

Las conjeturas de Pollock son dos hipótesis no probadas. Una de ellas es llamada “Conjetura de los números tetrahedros de Pollock”. Está descrita como: Todo entero positivo es la suma de a lo más cinco números tetrahedros. Un número tetrahedro es la suma de los primeros N números triangulares.

Un número triangular es la suma de los primeros N enteros positivos.

Los 1er, 2do, 3er y 4to números tetrahedros se pueden ver en la figura.



Es fácil mostrar que el N -ésimo número tetrahedro es igual a $\frac{N(N+1)(N+2)}{6}$. Por ejemplo, el cuarto número tetrahedro es $1 + (1 + 2) + (1 + 2 + 3) + (1 + 2 + 3 + 4) = 20$.

Asumiendo que la “Conjetura de los números tetrahedros de Pollock” es verdadera, tu programa debe calcular la mínima cantidad de números tetrahedros, tal que, la suma de ellos es igual al entero dado.

Input

La entrada empieza con un entero T que representa el número de casos de prueba. Las siguientes T líneas contienen un entero positivo X ($X \leq 10^6$).

Output

Para cada entero X , tu programa debe mostrar un entero que representa el número mínimo de números tetrahedros requeridos para que suma sea igual a X .

Example

Standard input	Standard output
2	2
14	2
5	

Explicación del caso 2:

$$5 = 4 + 1 = (1+3) + (1)$$

4 es el segundo número tetrahedro y 1 es el primero.

Problem G. Mr. Kuroki

Input file: Standard input
Output file: Standard output
Time limit: 2 second
Memory limit: 64 megabytes
Original idea: Nikolay Kuznetsov

Cierto día a Mr. Kuroki se le asignó la importante tarea de completar el proyecto final de su curso de algorítmica que constaba en realizar un programa de n líneas de código, el problema yacía en la fecha de entrega que era el día siguiente, por lo cual Mr. Kuroki decidió trabajar toda la noche. La tarea es extenuante y la forma de trabajo de Kuroki fue la siguiente:

Primero escribe v líneas de código, luego toma una taza de café, luego escribe $\lfloor \frac{v}{k} \rfloor$ líneas, toma otra taza de café, luego escribe $\lfloor \frac{v}{k^2} \rfloor$ líneas y así sucesivamente: $\lfloor \frac{v}{k^3} \rfloor$, $\lfloor \frac{v}{k^4} \rfloor$, $\lfloor \frac{v}{k^5} \rfloor$, ...

La expresión $\lfloor \frac{a}{b} \rfloor$ es tomada como la parte entera de dividir el número a entre el número b .

El momento en que el valor de $\lfloor \frac{v}{k^p} \rfloor$ sea 0, Mr. Kuroki caera profundamente dormido y solo despertará por la mañana, cuándo el programa ya deberá estar terminado.

Mr. Kuroki quiere saber, cuál será el mínimo valor que tomará v que le permitirá escribir **no menos** de n líneas de código antes de que caiga profundamente dormido.

Input

La primera línea contiene un entero T indicando el número de casos de prueba. En cada caso de prueba se darán dos enteros: N y K separados por un espacio — el tamaño del programa en líneas de código y el coeficiente de reducción de productividad, $1 \leq n \leq 10^9$, $2 \leq k \leq 10$.

Output

Imprimir un entero — el mínimo valor de v que permitirá a Mr. Kuroki completar el programa en una noche para cada caso.

Example

Standard input	Standard output
2	4
7 2	54
59 9	

NOTA: En el primer ejemplo el valor de $v = 4$. Mr. Kuroki escribe el programa siguiendo las siguientes porciones: primero 4 líneas, luego 2, y luego 1, y luego Mr. Kuroki cae dormido. Entonces, se las arregla para escribir $4 + 2 + 1 = 7$ líneas en una noche y completar su tarea.

En el segundo ejemplo la respuesta es $v = 54$. Mr. Kuroki escribe el programa siguiendo las siguientes porciones: 54, 6. que da la suma $54 + 6 = 60$, que es mayor a $n = 59$.

Problem H. El nuevo éxito de Michael Jackson

Input file: Standard input
Output file: Standard output
Time limit: 3 segundos
Memory limit: 64 megabytes
Original idea: Jhosimar Arias

Michael Jackson, el rey del pop, esta pensando en un título para su nueva canción. Desafortunadamente esta vez él no tiene mucha libertad para elegir el nombre, debido a que su manager quiere que el título sea un palíndromo. Un palíndromo es una secuencia de caracteres que leído en cualquier dirección, de derecha a izquierda y de izquierda a derecha, se obtiene la misma palabra. Por ejemplo: madam, racecar, aabaa, anitalavalatina, son todos palíndromos.

Un *palíndromo hermoso* es un palíndromo lexicográficamente mínimo que puede ser formado reordenando todos los caracteres de la cadena S . En para agradar a su manager, Michael Jackson tiene que encontrar un *palíndromo hermoso* que pueda ser formado con la cadena S . ¿Puedes ayudar a Michael con su nuevo éxito que batirá todos los récords?

Nota: La cadena $x_1x_2...x_p$ es lexicográficamente mas pequeña que la cadena $y_1y_2...y_q$, si se cumple cualquiera de las siguientes condiciones: $p < q$ y $x_1 = y_1, x_2 = y_2, ..., x_p = y_p$, o si existe un número r ($r < p, r < q$) tal que $x_1 = y_1, x_2 = y_2, ..., x_r = y_r$, y $x_{r+1} < y_{r+1}$.

Input

La primera línea de entrada contiene un entero T , el número de casos de prueba. Cada caso de prueba contiene una cadena no vacía S con a lo más 10^5 caracteres. Cada caracter de S es una letra minúscula del alfabeto inglés de la “a” a la “z”.

Output

Para cada caso de prueba, muestra el palíndromo lexicográficamente mínimo que pueda ser formado reordenando todas las letras de la cadena S . Si tal palíndromo no existe para una entrada dada, muestra -1.

Example

Standard input	Standard output
2 aabcc abc	acbca -1

Explicación

Para el primer caso, después de reordenar todos los caracteres, todos los palíndromos que pueden ser formados son *cabac* y *acbca*. De estos 2, el lexicográficamente mínimo es *acbca*.

Para el segundo caso no es posible formar un palíndromo con los caracteres dados.