Maratona de Programação de 2012

Instituto de Matemática e Estatística
Universidade de São Paulo

Caderno de Problemas

Departamento de Ciência da Computação IME-USP

Sábado, 18 de agosto de 2012.

Traducción al español por Jesús P. Mesias

Problema A: Festival das noites brancas

Arquivo: `brancas.[c|cpp|java]`

Todos los años, en la época de las llamadas "noches blancas" en el que el sol no se pone sobre la ciudad de Sao Petersburgo ocurre el "festival de las artes de las noches blancas", que consiste en una serie de presentaciones musicales, conciertos, bailes, y mucho más ,que atraen artistas de todo el mundo. Es considerado una de las mayores manifestaciones populares de toda Rusia, en el auge de las noches blancas,el festival acostumbra tener hasta un millón de participantes circulando por las calles de la ciudad. El Teatro Mariinski recibe algunos de los mejores espectáculos y, una vez que no tiene ingresos suficientes para todos los que desean asistir a las actuaciones, acostumbra utilizar un sistema curioso y divertido para sortear a los que podrán entrar en el teatro.

Cada persona que entra en el teatro, interesado en asistir a una presentación escoge una fila en la cual gustaría sentarse y recibe un ticket con un número de 000 a 999 escrito en el. Este número es el código del sorteo de aquella persona. Al llegar a la entrada el que atiende verifica la situación de la fila en la cual la persona se sentara. La fila es descrita por una secuencia de '1's y '0's, donde 1 indica asiento libre y 0 indica asiento ocupado. Esa secuencia es entonces interpretada como la representación binaria del número n. La persona entrara con sus acompañantes si el n-ésimo número de la secuencia de Fibonacci termina exactamente con el número escrito en su ticket. Así, por ejemplo, si la descripción de la fila es 100 la persona solo entrara si posee el ticket con el numero 003.

Entrada

La entrada es compuesta por diversas instancias. La primera línea de la entrada contiene un entero T indicando el número de instancias.

Cada instancia consiste en una linea que contiene una descripción de la fila. La descripción de una fila es una secuencia de '1's y '0's, nunca comenzando con '0' (el primer asiento de todas las filas estan todos reservados).

Salida

Para cada instancia imprima los 3 dígitos que deben estar escrito en el ticket para que la persona entre en el teatro.

Restricciones

Una fila no tiene más de 10,000 (diez mil) asientos.

Ejemplos

Exemplo de entrada	**Saída para o exemplo de entrada**
3 1 10 1010	001 001 055

Observación

Sea n un número entero positivo escrito en la base decimal. El n-ésimo número, $f(n)$, de la secuencia de Fibonacci es definido de la siguiente forma:

$$f(n) = \begin{cases} 1 & \text{si } n = 1 \text{ o } n = 2 \\ f(n-1) + f(n-2) & \text{caso contrario} \end{cases}$$

Problema B: Pontes de São Petersburgo

Arquivo: `pontes.[c|cpp|java]`

Todos conocen el famoso problema de los puentes de Königsberg, ciudad de Prússia que fue famosa por el problema resuelto por Euler en el siglo XVIII. Pocos conocen, mientras tanto, el problema de las puentes de San Petersburgo. La ciudad de San Petersburgo se localiza en las orillas del Rio Neva, y es cruzada por decenas de puentes que unían las orillas del rió con las centenas de pequeñas islas que el rió posee. Los moradores de la ciudad, conocedores del famoso problema de los puentes de Königsberg, crearon su propio problema. Los moradores saben que existen K puentes en la ciudad, que son R regiones distintas en la ciudad y que cada puente une exactamente 2 regiones distintas de la ciudad. Los moradores quieren saber si, para la ciudad de ellos, es posible escoger algunas de estas regiones tales que el número de puentes que incide en todas ellas es igual K. Note que, si dos de estas regiones escogidas tuvieran un puente entre ellas, este puente será contado dos veces.

Entrada

La entrada es compuesta por diversas instancias y termina con final de archivo(`EOF`).

La primera línea de cada caso de prueba contiene dos números, R y K, el número de regiones y puentes de la ciudad, respectivamente. Por efecto de simplificacion, las regiones son numeradas de 1 hasta R,inclusive. Siguiendo tenemos K líneas, cada una de ellas conteniendo dos números A y B, informando que existe un puente uniendo las regiones A y B de la ciudad.

Salida

Para cada caso de prueba imprima una línea solo con "S" (sin comillas), si es posible escoger las regiones de la manera descrita anteriormente, o "N"(sin comillas), si no es posible.

Restricciones

- $2 \leq R \leq 100$
- $1 \leq K \leq \frac{R \times (R-1)}{2}$

Ejemplos

Exemplo de entrada	**Saída para o exemplo de entrada**
2 1 1 2 3 3 1 2 1 3 3 2	S N

Problema C: Myachowski, o futebol russo

Arquivo: `futebol.[c|cpp|java]`

Muchos orígenes diferentes son atribuidos al fútbol. La actividad más antigua que se asemeja al fútbol era practicada en la China entre los siglos III y II a.C y se llamaba ts'uh Kúh (cuju),y consistía en jugar una pequeña pelota con los pies hacia una red. En Japón existe Kemari, practicado hasta hoy en eventos culturales. En Roma se jugaba el Harpastum, y en Grecia Epislcyros. Con el descubrimiento del nuevo mundo se descubrió también un juego Maya mucho semejante al futbol, el pok ta pok que tenia más de 3000 años de historia. En la edad media se jugaba en Florecia el Calcio Florentino, que muchos dicen que es la cuna del fútbol moderno. Hasta hoy indigenas del Amazonas juegan un juego muy semejante en que una pelota es empujada usando solo la cabeza en dirección a las metas enemigas. Sea como fuera, es casi imposible decir cual juego dio origen al fútbol jugado actualmente, cuyas reglas fueron formalmente establecidas por los ingleses en el final del siglo XIX.

Se tiene poca noticias de un juego ruso, también ancestral del fútbol y con reglas bastantes claras (como diría Arnaldo). Es Myachowski, también conocido como Otskok. El nombre viene probablemente de мяч (se lee myach) que significa "pelota" en ruso. En el juego un jugador entra en un campo que es una elipse cerrada y debe acertar en un hoyo localizado en la pared del campo.Sin embargo, el punto solo se calcula si la pelota entra en el hoyo después de ser pateada contra las paredes del campo, siendo desviada para dentro del hoyo.

Dada la posición inicial de la pelota, la dirección en la cual esta se moverá y la descripción del campo, su tarea es determinar los próximo dos puntos de contacto de la pelota con las paredes del campo. Considere que el centro del campo es la posición $(0, 0)$.

Entrada

La entrada es compuesta por diversas instancias y termina con final de archivo(`EOF`). Cada instancia consiste en una línea conteniendo 6 enteros,X, Y, D_x, D_y, A y B. La posicion inicial de la pelota es dada por el punto (X, Y) y la dirección por el vector (D_x, D_y). El campo tiene el formato de una elipse descrita por la ecuación.

$$\frac{x^2}{A^2} + \frac{y^2}{B^2} = 1.$$

Salida

Para cada instancia imprima dos líneas. La primera línea debe contener el primer punto de contacto de la pelota con la pared del campo y la segunda línea debe contener el segundo punto de contacto. Un punto de contacto debe ser impreso con dos números racionales separados por un espacio. Imprima los números con exactamente 3 decimales.

Restricciones

- $1 \leq A, B \leq 500$
- $-1000 \leq D_x, D_y \leq 1000$

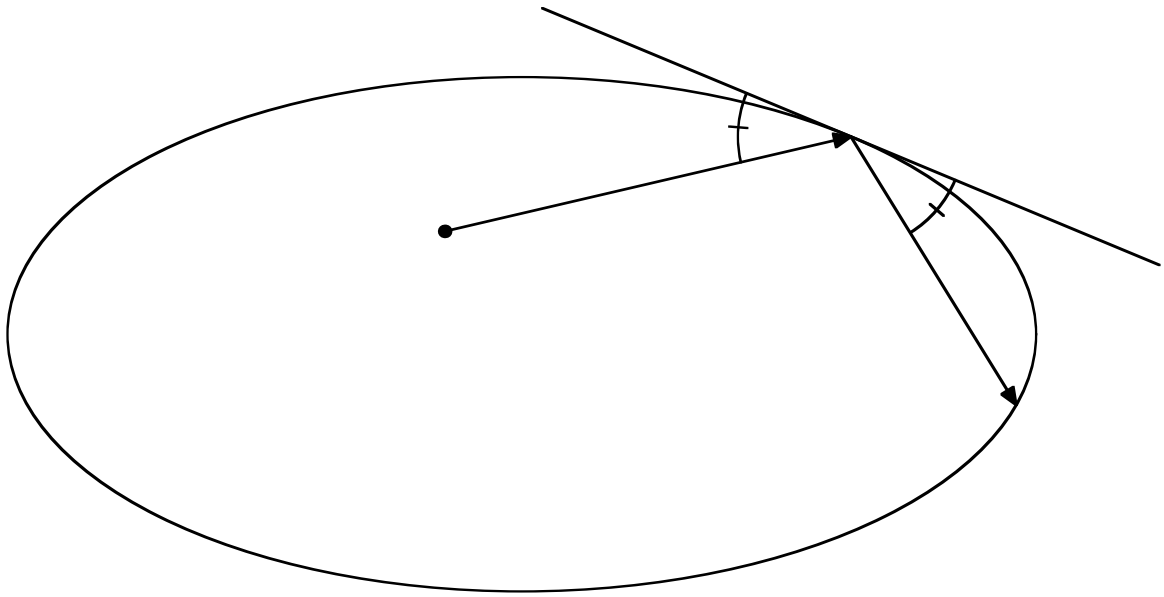

Figura 1: Angulo de entrada es igual al angulo de salida.

Ejemplos

Exemplo de entrada	**Saída para o exemplo de entrada**
0 0 1 1 1 1 0 0 1 0 2 3 -30 40 30 7 200 100	0.707 0.707 -0.707 -0.707 2.000 0.000 -2.000 0.000 127.955 76.856 192.310 -27.464

Rebote de la pelota

Suponga para ese problema que la pelota es reflejada perfectamente cuando golpea con la pared del campo. Eso es, el ángulo que el vector de entrada hace con la recta tangente a la pared del campo en el punto de contacto es el mismo que el ángulo de salida. La figura 1 ejemplifica ese comportamiento

Problema D: Cerco a Leningrado

Arquivo: `cerco.[c|cpp|java]`

La ciudad de San Petersburgo cambio de nombre después de la revolución rusa en 1914 a Petrogrado. Después de la muerte de Lenin, en homenaje a el grande líder el nombre de la ciudad cambio nuevamente para Leningrado en 1924, y así permanecio hasta el fin de la Unión Sovietica. En 1991, la ciudad volvió a tener el nombre antiguo. Durante la segunda guerra mundial la ciudad de Leningrado sufrió un cerco de las tropas alemanas que duro cerca de 900 días. Fue una época terrible, de mucha hambre y perdidas humanas, que termino el 27 de enero de 1944 con la victoria de los soviéticos. Es considerada una de las victorias más costosas de la historia en términos de perdidas de vidas humanas.

En el auge de la ofensiva alemán, en el año de 1942, varios francotiradores de elite fueron esparcidos por la ciudad, inclusive, en algunos puntos estrategicos de la ciudad más de un tirador esperaba a los soldados enemigos. El espionaje ruso tenia información detallada de las habilidades de esos francotiradores, pero sus escondites eran excelentes, convirtiendo la tarea de un soldado soviético que desee cruzar la ciudad en una tarea extremadamente difícil. Los soldados soviéticos estaban muy bien entrenados, pero con el pasar del tiempo y la continuación del cerco de la ciudad, los mejores soldados fueron siendo diezmados, una vez que se equivocaron en el primer intento eran asesinados por los soldados alemanes en la emboscada.

Sabiendo la probabilidad de un soldado en matar un francotirador alemán y sabiendo también el número de balas que el tenia en su disposición , deseamos saber la probabilidad de éxito del soldado en llegar a un punto estratégico destino, partiendo de un punto estratégico de origen. El soldado , siendo muy experimentado, siempre usaba un camino que maximizaba la probabilidad de éxito. Note que el soldado debe matar todos los francotiradores presentes en el camino usado, inclusive los que estarían en puntos estratégicos de origen y destino.

Entrada

La entrada es compuesta por diversas instancias y termina con final de archivo(`EOF`).

La prima línea de cada instancia contiene 3 enteros,N, M, y K y la probabilidad P del soldado para matar a un francotirador. Los enteros N, M, y K representan respectivamente los números de puntos estrategicos , vías uniendo puntos estrategicos y balas cargadas por el soldado soviético. Los puntos estrategicos son enumerados de 1 a N.

Cada una de las próximas M líneas contienen un par de enteros i y j indicando que existe una via uniendo el punto i a el j. En seguida tiene una línea conteniendo un entero A, correspondiendo a el numero de francotiradores en la ciudad, seguido por A enteros indicando la posición de cada francotirador.

La última línea de cada instancia contiene dos enteros indicando el punto de partida y el destino del soldado.

Salida

Para cada instancia imprima una línea conteniendo la probabilidad de éxito del soldado soviético. La probabilidad debe tener una aproximación de 3 dígitos.

Restricciones

- $2 \leq N \leq 1000$
- $0 \leq K \leq 1000$
- $0 \leq A \leq 2000$
- $0 \leq P \leq 1$

Ejemplos

Exemplo de entrada	Saída para o exemplo de entrada
3 2 10 0.1 1 2 2 3 10 1 1 3 3 1 3 1 1 3 3 1 3 5 5 10 0.3 1 2 2 4 2 5 4 5 5 3 6 3 3 3 3 3 3 1 3	0.000 0.001

Problema E: Desafio de São Petersburgo

Arquivo: `desafio.[c|cpp|java]`

Rusia siempre fue cuna de grandes maestros de ajedrez. Pocos saben, pero la FIDE (Federacion Internacional de Ajedrez), que es el órgano máximo regulador del juego de ajedrez fue fundada en 1924, a partir de un movimiento iniciado 10 años antes en el campeonato mundial de la modalidad ocurrida en San Petersburgo en 1914. Hoy, entre los 10 mejores del mundo, según la FIDE, 3 son rusos.

El torneo de San Petersburgo fue conocido también por los intentos de los grandes maestros de popularizar el juego. En la época los grandes maestros (como Capablanca) fueron por las calles proponiendo desafíos para las personas, con el objetivo de interesarlas a practicar el juego. Uno de esos desafíos fue conocido como el desafió de San Petersburgo. El gran maestro armaba una situación en que las piezas blancas tenían solo al Rey, y el objetivo era que la persona diga si el Rey Blanco estaba o no en Jaque Mate.

En la situación descrita arriba, el Rey Blanco esta en Jaque Mate si el esta siendo atacado y cualquier movimiento que el haga en una nueva casilla también este siendo atacada.

Entrada

La entrada es compuesta por diversas instancias y termina con el final de archivo.

La primera línea de cada instancia contiene un entero N indicando el número de piezas negras. La línea siguiente contiene la descripción de las N posiciones de las piezas negras separadas por un espacio. La tercera línea contiene una descripción, la del Rey Blanco.

Una descripción de una pieza consiste de 3 caracteres. El primero indica si la pieza es un Peón (P) , Torre (T), Alfil (B), Reina (R) o Rey (W). Note que el gran maestro no usaba caballos para facilitar a las personas que aun estaban comenzando a aprender el juego. El segundo carácter, entre 'a' y 'h', indica la columna en la cual la pieza esta y el tercero, de '1' a '8' indica la fila.

En ninguna de las instancias ocurre la situación en la que el Rey Blanco y el Rey Negro son adyacentes.

Salida

Para cada instancia, imprima una línea con la palabra `SIM`,si el Rey Blanco esta en Jaque Mate, o la palabra `NAO`,caso contrario.

Restricciones

- $2 \leq N \leq 10$

Ejemplos

Exemplo de entrada	**Saída para o exemplo de entrada**
4 Tc3 Te4 Bf4 Wa8 Wd1 5 Wb5 Pf3 Rh7 Te1 Tg1 Wh1 4 Wa8 Ta1 Ta3 Rd2 We2	NAO SIM NAO

Lo que necesita saber sobre el Ajedrez

Considere que inicialmente las piezas del jugador negro están en las filas 7 y 8 mientras que la del jugador blanco inician en las filas 1 y 2. No puede haber dos piezas en la misma casilla. Las piezas consideradas en el problema (peón, torre, alfil , reina y rey) no pueden pasar por encima de otras piezas, es decir, si durante su movimiento existe alguna pieza en su camino usted debe parar antes o atacar a la pieza(si ella fuera del oponente), tomando ese lugar. El movimiento y el ataque de cada tipo de pieza se dan de la siguiente forma:

- Peón: se mueve solo una casilla para el frente (en dirección a la fila 1), con posibilidad de atacar en cualquiera de una de las dos casillas diagonales inmediatas a su frente.
- Torre: mueve/ataca cuantas casilla quisiera o en la horizontal, o en la vertical;
- Alfil: mueve/ataca cuantas casillas quisiera en la diagonal;
- Reina: mueve/ataca cuantas casillas quisiera o en la horizontal, o en la vertical, o en diagonal;
- Rey: mueve/ataca solo una casilla o en la horizontal, o en la vertical, o en diagonal.

Problema F: Os benefícios da vodka

Arquivo: `beneficios.[c|cpp|java]`

San Petersburgo es conocida como la capital de la cerveza rusa y abriga diversas cervezerias importantes. Dicen que la calidad de agua de la ciudad es la responsable por una cerveza de excelente calidad. Además de fabricas tradicionales, como Heineken, algunas marcas locales son destacadas, como a Tinkoff y Baltika. También en la ciudad son producidas algunos de los mejores vodkas del mundo. El más antiguo, llamado Liviz, data de 1897. Esta destilería produce vodkas de excelente calidad, medida por padrones internacionales. Curiosamente, algunos tipos de vodkas, cuando son consumidos juntos, acaban teniendo, según los especialistas, un sabor mucho mejor. De esa forma, algunos tipos de vodka son reunidos en categorías que, cuando son compradas totalmente por el consumidor, traen un beneficio agregado según padrones internacionales de calidad. Cada uno de los vodkas tienen un precio asociado, y su tarea es encontrar una compra que maximice el beneficio total menos el costo de los vodkas adquiridos.

Trascribiendo, cada vodka tiene un costo C_j y existen M categorías diferentes, cada cual con un beneficio B_i. Un beneficio solo es calculado si todos los tipos de vodka que componen la categoría son adquiridos. Una misma botella de vodka puede participar de más de una categoría para calcular el beneficio. Su tarea es determinar cuales tipos de vodka comprar de forma que se maximice la suma de los beneficios adquiridos menos el costo de los items comprados. Usted puede suponer que fue a Rusia con dinero suficiente para comprar todos los tipos de vodka producidos por Liviz (Buen provecho :D)

Entrada

La entrada es compuesta por diversas instancias y termina con final de archivo(`EOF`). La primera línea de cada instancia contiene dos enteros N y M representando, respectivamente, la cantidad de tipos diferentes de vodka en venta y el número de categorías existentes. Los tipos de vodka son identificados por números de 1 a N y las categorías por números de 1 a M.

La línea siguiente contiene N enteros, C_j, separados por espacio, correspondiendo al costo del vodka j. En la siguiente línea existen M enteros, P_i, separados por espacios, indicando cuantos tipos diferentes de vodkas componen la categoría i.

Cada una de las M líneas siguientes describe una categoría comenzando con un entero, B_i indicando su beneficio, seguido por los tipos de vodka que la compone, separados por espacios.

Salida

Para cada instancia imprima, en una unica línea, el mayor valor que puede ser obtenido por la suma de los beneficios de las categorías adquiridas menos el costo de los tipos de vodkas comprados.

Restricciones

- $1 \leq N \leq 600$
- $1 \leq M \leq 400$
- $1 \leq C_j \leq 1000$ para $1 \leq j \leq N$
- $1 \leq P_i \leq N$ para $1 \leq i \leq M$
- $1 \leq B_i \leq 1000$ para $1 \leq i \leq M$

Ejemplos

Exemplo de entrada	Saída para o exemplo de entrada
2 3 80 80 1 2 1 90 1 50 1 2 25 2 4 3 50 200 50 130 2 2 2 70 1 2 260 2 3 120 3 4	10 30

Observaciones

En el primer ejemplo el valor máximo puede ser obtenido comprando solo el vodka 1 y adquiriendo la categoría 1. En el segundo ejemplo el valor máximo puede ser obtenido comprando los vodkas 1, 2 y 3 y adquiriendo las categorías 1 y 2.

Problema G: As dinastias de São Petersburgo

Arquivo: `dinastias.[c|cpp|java]`

San Petersburgo fue fundada el día 27 de mayo de 1703 por Pedro, el Grande, y fue capital imperial de la Rusia por un periodo corto luego después(1713 a 1728) y luego casi dos siglos, desde 1732 hasta 1918. En este último período el trono imperial ruso acabo siendo ocupado por diversos emperadores,muchas veces de líneas de dinastías diferentes. En la traducción imperial rusa se llama текущий(se lee текущий [1]) una secuencia de descendientes dentro de una dinastía, osea, un elemento, su hijo, su nieto, y así para adelante. La determinación de estas текущий es fundamental cuando se desea determinar el sucesor del actual emperador, una vez que el próximo emperador es el elemento vivo de una текущий que este más de próxima del actual emperador. Es claro que un árbol genealógico puede ser dividida en текущий de varias formas diferentes. Lo interesante es encontrar una coparticipación que minimice el número de текущий necesario para cubrir todos los elementos de la dinastía.

Su tarea en este problema es determinar, dada el árbol genealógico de la familia imperial rusa, el menor número de текущий que particionan toda la familia imperial, eso es, todos los emperadores tienen que pertenecer a exactamente un текущий y esos tienen que ser el menor número posible.

Entrada

La entrada es compuesta por diversas instancias y termina con final de archivo(`EOF`). La primera línea de cada instancia contiene dos enteros N y M representando, respectivamente, la cantidad de emperadores y el número de relaciones de filiación en aquella instancia. Los emperadores son identificados por números de 1 a N. Cada una de las próximas M líneas contienen dos enteros P_i y F_i,indicando que P_i es padre de F_i. Una particularidad del árbol genealógico dado es que en caso de dudas de paternidad, todos los posibles padres eran indicados, osea, una persona puede tener cualquier número de padres.

Salida

Para cada instancia imprima una línea conteniendo un único número entero, que es el número mínimo de текущий necesarios para particionar todos los emperadores de aquella instancia.

Restricciones

- $1 \leq N \leq 1{,}000$
- $0 \leq M \leq 10{,}000$
- $1 \leq P_i < F_i \leq N$

[1] Ruso es una lengua fonética ;)

Ejemplos

Exemplo de entrada	Saída para o exemplo de entrada
3 2 1 2 2 3 3 2 1 3 2 3 5 4 1 3 2 3 3 4 3 5 4 4 1 2 1 3 1 4 2 4	1 2 3 2

Problema H: Festas de São Petersburgo

Arquivo: `festas.[c|cpp|java]`

San Petersburgo se torno después del fin de la cortina de hierro, en el inicio de los años 90, una de las principales ciudades de la escena alternativa en todo el mundo. Grupos de punks, diversas bandas de hardcore y otros representantes de la escena alternativa se mudaron para la ciudad, atraídos por la grande cantidad de jóvenes. Con el surgimiento de las comunidades virtuales, algunos años más tarde, se noto el enorme potencial del uso de estas comunidades para coordinar encuentros, fiestas, raves, etc. En estas fiestas de San Petersburgo es siempre muy importante que cada uno de los participantes tenga por lo menos un cierto número de amigos en la red social. Y, al mismo tiempo, deseamos invitar al mayor número posible de personas de San Petersburgo que cumplan con la restricción. Tal restricción dice que, para ser invitada a la fiesta, la persona necesita tener por lo menos un número K de amigos en la lista de invitados.

Su tarea en este problema es, dado el conjunto de personas de la comunidad y la lista de sus relaciones, determinar cuales deben ser llamadas para que la fiesta tenga la mayor cantidad posible de participantes satisfaciendo tal restricción.

Entrada

La entrada es compuesta por diversas instancias y termina con final de archivo(`EOF`).

La primera línea de cada instancia contiene tres enteros N, M y K representando respectivamente el número de personas en la comunidad, el número de relaciones de amistad en esa comunidad y el número minimo de amigos invitados que una persona necesita tener para ser invitada. Cada persona de la comunidad es identificada por números de 1 a N. Cada una de las siguientes M líneas contienen un par de personas indicando que ellas son amigas en la red social.

Salida

Para cada instancia imprima una única línea conteniendo la lista de las personas a ser invitadas separadas por un espacio en blanco. La lista debe estar ordenada en orden creciente. Caso nadie pueda ser invitado, imprima el número 0.

Restricciones

- $1 \leq N \leq 1000$
- $0 \leq K \leq N$

Ejemplos

Exemplo de entrada	Saída para o exemplo de entrada
6 6 2 1 3 3 5 2 3 2 4 4 6 6 2 6 6 3 1 2 2 3 3 1 4 5 5 6 6 4	2 4 6 0

Problema I: Produção ótima de ótima vodka

Arquivo: `vodka.[c|cpp|java]`

La producción de vodka de la ciudad de San Petersburgo es famosa en todo el mundo. Cuenta la leyenda que el vodka producida es distribuida directamente en la casa de algunos de los funcionarios más importantes de la empresa a través del sistema de abastecimiento de agua. Es decir, basta abrir el grifo y el vodka saldrá helado (al final los caños están corriendo a una temperatura negativa en la mayor parte del año) del caños. Esto causa diversos problemas de seguridad, al final las personas secaban las calles buscando los supuestos caños de vodka saliendo de la empresa.

Este no es el único problema que se enfrenta en la producción del vodka de la ciudad. Para garantizar el padrón de calidad exigido de la bebida, ella es producida en solo un destilador, que tiene una vida útil bien definida, de M años. Su mantenimiento depende de la edad del equipo. El costo de mantenimiento es C_i, donde i es la edad del destilador, y debe ser pagado cada año, hasta para los destiladores nuevos. Estos destiladores tienen un precio P cuando son comprados nuevos (edad 0) y los destiladores usados en fábricas rusas son disputados por destilerías del todo el mundo, donde son usados por muchos años, y por museos. El precio de venta de un destilador con edad i es V_i. Note que un destilador con edad M no puede seguir siendo usado y debe ser vendido.

Su tarea en este problema es decidir cuales instantes la empresa deberá cambiar el destilador de forma de minimizar el costo de producción al final de N años (a partir del año 1). Considere que el cambio de destiladores solo puede ser efectiva al inicio de año.

Entrada

La entrada es compuesta por diversas instancias y termina con final de archivo(`EOF`).

La primera línea de cada instancia posee 4 enteros,N, I, M y P representando, respectivamente, el periodo de producción, la edad inicial del destilador , la edad máxima del destilador y el precio de un destilador nuevo.

La línea siguiente contiene M enteros, separados por espacios, correspondiendo al costo de mantenimiento C_i, para $i = 0, 1, 2, \ldots, M-1$. La próxima y última línea contiene M enteros, separados por espacios, correspondiendo al valor de venta V_i, para $i = 1, 2, \ldots, M$.

Salida

Para cada instancia la salida debe contener dos líneas. En la primera, imprima el costo mínimo para el periodo dado. En la segunda, una secuencia creciente de enteros, separados por espacios, indicando los años en los cuales son cambiadas las máquinas. Si la máquina nunca es cambiada, entonces imprima solo un 0. Caso exista más de una secuencia posible, escoja aquella en la cual las máquinas son cambiadas lo más temprano posible y siempre que sea posible(por ejemplo, entre las secuencias “1 4 7” y “1 2 8 10 14” escoja la segunda).

Restricciones

- $1 \leq N \leq 2{,}000$
- $1 \leq M \leq 2{,}000$
- $1 \leq I \leq M$
- $1 \leq P \leq 1{,}000$
- $1 \leq C_i \leq 1{,}000$
- $1 \leq V_i \leq P$

Ejemplos

Exemplo de entrada	Saída para o exemplo de entrada
4 2 6 100 30 50 65 80 100 120 60 50 40 30 20 10 5 5 6 200 1 100 100 100 100 200 50 100 100 100 100 100	260 1 3 501 1

Problema J: Lista telefônica econômica

Arquivo: `lista.[c|cpp|java]`

Debido al gran número de reclamaciones, la compañía telefónica de San Petersburgo esta siendo obligada a invertir más en la mejora de sus servicios. Para eso la compañía decidió disminuir el presupuesto de algunos sectores para aumentar el de otros más esenciales. Uno de los sectores que tendrá su presupuesto reducido es el de impresión de listas telefónicas.

Con un presupuesto reducido, el sector de impresión de listas telefónicas no consigue comprar toner suficiente para imprimir las listas completas. Como los números de teléfonos son impresos alineados en la vertical, fue sugerida la siguiente solución: a partir del segundo número de teléfono impreso, los dígitos iniciales del próximo número a ser impreso deben coincidir con los números de arriba que son omitidos, quedando solo un espacio en blanco. Por ejemplo, para los números 535456, 535488, 536566 y 835456 la impresión es la siguiente:

```
5 3 5 4 5 6
        8 8
    6 5 6 6
8 3 5 4 5 6
```

Note que esta impresión economizo la impresión de 6 caracteres. La compañía telefónica cuestiono también no imprimir los sufijos repetidos, pero en las pruebas hechas cambiaron que la respuesta no fue buena para el usuario y decidieron, por tanto, hacer solo la eliminacion de los prefijos. Para saber si la economía sera suficiente, el sector de impresión quiere saber el número máximo de caracteres que pueden ser omitidos. No en tanto, como en cualquier ciudad grande, son varios los números telefónicos y ellos no quieren gastar hombres-hora para calcular manualmente este valor. Entonces usted, nuevo empleado de la compañía telefónica, automatizar la economía hecha por el toner, en el número de caracteres.

Entrada

La entrada es compuesta por diversas instancias y termina con final de archivo(`EOF`).

Cada caso de prueba contiene un entero N, que informa el número de teléfonos den la lista. Las próximas N líneas poseen, cada una de ellas , un teléfono X_i, de hasta 20 caracteres. Para un mismo caso de prueba los números de teléfono tienen la misma cantidad de caracteres. Un número de teléfono puede comenzar con el carácter '0'.

Salida

Para cada caso de prueba imprima una línea informando el mayor número posible de caracteres economizados por este proceso.

Restricciones

- $1 \leq N \leq 10^5$

Ejemplos

Exemplo de entrada	**Saída para o exemplo de entrada**
2 12345 12354 3 535456 535488 835456	3 4