



Concurso de Programación AdaByron 2020

<http://www.ada-byron.es>

Cuadernillo de problemas

Patrocinado por


accenture

ThoughtWorks®



Realizado en la **Escuela Técnica Superior de Ingeniería Informática (URJC)**
28-29 de febrero de 2020

 Universidad
Rey Juan Carlos

In almost every computation a great variety of arrangements for the succession of the processes is possible, and various considerations must influence the selections amongst them for the purposes of a calculating engine. One essential object is to choose that arrangement which shall tend to reduce to a minimum the time necessary for completing the calculation.

Ada Byron

Listado de problemas

A Enero: Dickie está en aprietos	3
B Febrero: En el último minuto	5
C Marzo: La moda secundaria	7
D Abril: Declaración de la renta	9
E Mayo: Dickie y las hamburguesas	11
F Junio: Rebajas de Verano	13
G Julio: Torneos de Ajedrez - Reto Accenture	15
H Agosto: HH y el Bosón	17
I Septiembre: Vuelta a la rutina	19
J Octubre: Recogiendo caramelos	21
K Noviembre: La casa encantada	23
L Diciembre: Dora la Aisladora	25

Autores de los problemas:

- Antonio González Pardo (Universidad Rey Juan Carlos)
- Isaac Lozano Osorio (Universidad Rey Juan Carlos)
- Jakub Jan Luczyn (Universidad Rey Juan Carlos)
- Jesús Sánchez-Oro Calvo (Universidad Rey Juan Carlos)
- Raúl Martín Santamaría (Universidad Rey Juan Carlos)
- Sergio Pérez Peló (Universidad Rey Juan Carlos)
- Dixon David Morán González (ThoughtWorks)
- Daniel Ampuero Anca (Amazon)
- Leonardo Antonio Santella Dambrosio (Amazon)
- Héctor Enrique Navarro Urbina (Amazon)

● A

Enero: Dickie está en aprietos

Dickie tiene una cita para renovar su residencia. ¡Él debe llegar puntualmente a la oficina de inmigración o no podrá renovar su residencia! No obstante, hay una terrible nevada cayendo sobre Madrid y puede que algunas calles estén cerradas.

Para asegurarse de que llegará a tiempo a su cita, Dickie ha reservado un Kart2Good por exactamente K minutos. Dado que es una persona austera, Dickie no quiere usar su Kart2Good por menos o más minutos de los que lo ha reservado.

Pero Dickie tiene una carta bajo su manga: tiene muchos amigos con los que pernoctar y reducir la posibilidad de perder su cita. Dickie tiene M amigos y sus casas están ubicadas en varias localizaciones de Madrid.

La ciudad está representada por una matriz $N \times N$, donde cada posición (i, j) representa la probabilidad (de 0 a 100) de que Dickie coja la calle que va del punto i al punto j . Si (i, j) es 0, significa que no hay probabilidad alguna de que Dickie coja esa calle. Se tarda 1 minuto en recorrer cada calle.

¿Podrías ayudar a Dickie a escoger con quién debería pernoctar?

Entrada

La primera línea contiene un entero T que especifica el número de casos de prueba. A continuación, habrá T casos de prueba con la entrada descrita a continuación.

Cada caso comenzará con tres enteros: N , M y K ; los cuales representan el tamaño de la ciudad, la cantidad de amigos de Dickie y el tiempo que ha reservado el Kart2Good.

A continuación habrá N líneas con N números cada una representando una matriz con las probabilidades mencionadas anteriormente.

Seguidamente, una línea con M enteros y las posiciones de cada uno de los amigos de Dickie en Madrid.

Salida

Por cada caso i , se deben imprimir M líneas, cada una con dos números M_i y P_i , representando la ubicación de un amigo de Dickie y la probabilidad de llegar en exactamente K minutos partiendo de esa ubicación. Las ubicaciones deben estar ordenadas de menor a mayor. P_i debe ser impreso con 15 dígitos de precisión. Se considerarán válidas las respuestas que presenten una diferencia absoluta o relativa menor o igual a $10e - 6$.

Si no hay ninguna probabilidad de que Dickie llegue a su cita desde ninguna de las casas de sus amigos, imprimir una línea con "Hasta la vista, chaval!".

Entrada de ejemplo

```
3
4 2 100
0 50 50 0
50 0 0 50
10 0 0 90
0 0 30 70
1 2
4 1 100
0 100 0 0
100 0 0 0
10 0 0 90
0 0 30 70
1
4 3 100
100 0 0 0
0 100 0 0
0 0 100 0
0 0 0 100
1 2 3
```

Salida de ejemplo

```
1 0.721649484536080
2 0.721649484536080
Hasta la vista, chaval!
Hasta la vista, chaval!
```

Límites

- $2 \leq N \leq 100$
- $1 \leq M \leq N$
- $K \leq 5000$
- $\sum_{j=1}^N P_{i,j}=1$
- Tiempo: 4 segundos

● B

Febrero: En el último minuto

De nuevo llega febrero y, con él, San Valentín. Como siempre, a David se le ha olvidado comprar un regalo para Manuela y tiene que salir a comprar uno en el último minuto. Su objetivo es poder dárselo cuando ella vuelva del trabajo, por lo que no tiene mucho tiempo. David sabe que Manuela se enfadará mucho si no tiene su regalo, y con razón, ya que ella siempre hace unos regalos espectaculares a David.

Dado que, un año más, David va a contrarreloj para encontrar el regalo perfecto, no puede visitar todas las tiendas que le gustaría, así que ha seleccionado N tiendas en las que buscará regalos que puedan sorprender a Manuela. Cada tienda se encuentra a una distancia D de su casa.

Como Manuela siempre prepara unos regalos alucinantes, David quiere asegurarse de estar a la altura, por lo que quiere visitar todas las tiendas que ha preseleccionado antes de comprar el regalo.

De nuevo, vamos a ser buenos amigos y vamos a ayudar a David a encontrar la ruta más corta de forma que visite todas las tiendas y llegue a tiempo de vuelta a casa para que Manuela no sospeche que su regalo ha sido adquirido en el último minuto.

Entrada

La primera línea contiene un entero T que indica el número de casos de prueba. A continuación, se especifican T casos de prueba.

Cada caso comenzará con un número entero N indicando la cantidad de tiendas que ha seleccionado David. A continuación, vendrán N líneas con $N_{i,j}$ enteros, indicando la distancia desde la tienda i a la tienda j . Consideramos que la casa de David siempre será $N = 0$.

Salida

Por cada caso se debe imprimir la longitud de la ruta más corta desde la casa de David pasando por todas las tiendas y regresando a casa antes de que vuelva Manuela.

Entrada de ejemplo

```
1
4
0 10 15 20
10 0 35 25
15 35 0 30
20 25 30 0
```

Salida de ejemplo

```
80
```

Límites

- $2 \leq N \leq 15$
- $1 \leq D \leq 1000$
- Tiempo: 3 segundos



Marzo: La moda secundaria

Es el mes de Marzo y la empresa de Yoyo Armando se prepara para un aumento en sus ventas debido a su gran catálogo de prendas de vestir y su calidad en las mismas. Sin embargo, el grupo Yoyo ha decidido optar por su siempre confiable grupo de especialistas que votarán qué tipos de prendas de vestir tendrán más éxito en el mercado. Por ejemplo, “los mocasines Yoyo first edition” parece que tendrán algo de éxito esta temporada.

Por un cambio de mánager, parece que la empresa de Yoyo Armando no está interesada en saber qué es lo que sus especialistas votan más, sino cuál es el i -ésimo conjunto de prendas más votadas en su lugar, es decir, el grupo de Yoyo está interesado en saber cuáles son las prendas que cumplen ser la i -ésima moda de todo el conjunto.

Entrada

Un entero N denotando el número de prendas, luego, N líneas con el nombre de las prendas (con mayúsculas, minúsculas, espacios y de hasta 30 caracteres de longitud). Finalmente, un entero K denotando la K -ésima moda que el grupo Yoyo quiere saber .

Salida

La salida tendrá un número P denotando el número de prendas diferentes que cumplen con ser la K -ésima moda y luego P líneas, una por cada producto, los nombres deben ir ordenados en orden de aparición como lo hicieron en la descripción de la entrada. Si no hay ninguna moda que cumpla las descripciones solo se imprime un 0.

Entrada de ejemplo

```
10
Mocasines Yoyo Armando
Camisa ElCoust
Calcetines Marinela
Calcetines Marinela
Mocasines Yoyo Armando
Camisa ElCoust
Mocasines Yoyo Armando
Mocasines Yoyo Armando
Calcetines Marinela
Camisa ElCoust
2
```

Salida de ejemplo

```
2
Camisa ElCoust
Calcetines Marinela
```

Límites

- $1 \leq N \leq 40,000$

- Tiempo: 0.8 segundos

● D

Abril: Declaración de la renta

Los primeros días de abril se suele hacer la declaración de la renta (tributo que tienen que pagar todos los ciudadanos a la Agencia Tributaria, en relación con los ingresos que hayan obtenido durante un año). Los ingenieros del ministerio quieren crear un programa que les ayude con los diferentes nombres de los ciudadanos. A la hora de rellenar el nombre en la declaración, la gente que tiene nombre compuesto a veces duda qué nombre poner o pueden existir errores tipográficos que dificultan los procesos automáticos de la declaración de la renta. Para facilitar esta tarea, necesitan que, dado un nombre, se indique si es un nombre *parecido*. Un nombre se considerará *parecido* si:

- Es prefijo de uno de los nombres anteriores o,
- Uno de los nombres ya utilizados es prefijo suyo.

Entrada

La entrada tendrá un número T denotando el número de casos que vienen a continuación. Seguidamente vendrá cada uno de estos casos, los cuales contendrán un número M denotando el número de nombres a comparar. A continuación de este M vendrán los M nombres de tamaño N a comparar, de los que se tienen que obtener los parecidos. Todos ellos son letras minúsculas del alfabeto inglés y sin espacios.

Salida

La salida contendrá un número mostrando el total de nombres parecidos y, a continuación, se mostrará el listado de los nombres parecidos.

Entrada de ejemplo

```
1
4
marco
ma
marc
matias
```

Salida de ejemplo

```
3
ma
marc
matias
```

Límites

- $1 \leq T \leq 10$
- $1 \leq M \leq 10000$
- $1 \leq N \leq 500$
- Tiempo: 1 segundo



Mayo: Dickie y las hamburguesas

Todo sabemos que Dickie es un amante de las hamburguesas. Sin embargo, se ha vuelto un poco selectivo en este ámbito.

Dadas N hamburguesas, las cuales tienen un valor v_i y un peso w_i cada una, queremos ayudar a Dickie a hallar el conjunto de hamburguesas, de tamaño k , tal que la felicidad de Dickie sea la máxima posible. . . Sí, se ha vuelto muy selectivo.

Para Dickie, la felicidad que le aporta un conjunto de hamburguesas $S = \{h_1, h_2, \dots, h_k\}$, es representada por la siguiente función matemática:

$$f(S) = \frac{\sum_{i=1}^k v_i}{\sum_{i=1}^k w_i}$$

Entrada

La primera línea de la entrada contiene 2 enteros, n , que especifica el número de hamburguesas, y el número k , que define la cardinalidad del conjunto de hamburguesas requerido.

Seguidamente, n líneas especificando para cada hamburguesa el valor v_i y el peso w_i .

Salida

La salida esperada es la felicidad máxima posible. Se espera, además, que la salida tenga 6 decimales de precisión.

Entrada de ejemplo

```
3 2
1 2
1 3
1 1
```

Salida de ejemplo

```
0.666667
```

Límites

- $1 \leq K \leq N \leq 10^5$
- $0 \leq v_i \leq 10^6$
- $0 \leq w_i \leq 10^6$
- Tiempo: 1 segundo



Junio: Rebajas de Verano

Las vacaciones de verano ya están cerca. Gaben, gracias a su experiencia en las competiciones de programación, ha conseguido aprobar todas las asignaturas este año y está a punto de disfrutar de unas largas vacaciones. Quiere aprovechar este descanso para jugar a todos los videojuegos que ha ido guardando a lo largo del curso en su lista de deseos. Esta lista, por una manía que tiene, siempre contiene una cantidad de juegos que pueda representarse como potencia de dos.

Parece que la suerte está de su parte porque su plataforma favorita *Vapour* acaba de lanzar una oferta especial, posiblemente debido a que muchos de sus clientes se están trasladando a la plataforma competidora *Awesome Store*. La oferta permite comprar la mitad de la lista de deseos de un usuario con un descuento considerable.

Parece que la oferta puede aplicarse varias veces, mientras se tenga más de un juego en la lista de deseos. Así que el usuario puede comprar la mitad de su lista y, luego, volver a aplicarla para comprar la mitad de lo que ha quedado, así repetidamente. El único problema es que no se puede cambiar el orden de los juegos en la lista.

Ya que tiene mucho tiempo libre, Gaben ha decidido aprovechar la oferta para comprar todos los juegos de su lista de deseos, pero quiere gastarse lo menos posible. Al aplicar la oferta solamente puede elegir si quiere comprar los primeros $N/2$ juegos de su lista o los $N/2$ últimos, siendo N la cantidad de juegos que tiene en cada momento. No es un problema muy complicado, pero necesitará pensarlo bien para no pagar de más.

Entrada

La entrada contendrá varios casos de prueba T , indicados en la primera línea. Cada caso de prueba constará de dos líneas, en la primera vendrán dos números, N , indicando el tamaño de la lista de deseos, y D , el descuento, con valor entero, en tanto por ciento (%) que se aplica con la oferta. En la segunda línea vendrán los precios (P_i), con dos cifras decimales, de los videojuegos en el orden en el que aparecen en la lista de deseos de Gaben, separados por espacios.

Salida

Por cada caso de prueba, imprimir en una línea el precio mínimo a pagar por comprar todos los videojuegos de la lista de deseos, aprovechando al máximo la oferta, con dos cifras decimales redondeadas.

Entrada de ejemplo

```
2
16 15
27.64 12.49 88.85 99.71 75.26 15.06 69.19 55.20 80.94 71.83 75.03 52.77 22.93 44.77
80.33 84.57
16 46
87.44 22.10 14.01 78.64 72.27 29.14 12.44 52.42 60.31 74.59 73.88 82.47 67.31 7.57
24.31 10.86
```

NOTA: El listado de precios aparece en varias líneas por cuestiones de formato del problemario, la entrada del juez sigue la descripción proporcionada.

Salida de ejemplo

```
814.96
419.15
```

Límites

- $1 \leq N \leq 1048576$
- $0 \leq D \leq 100$
- $0.01 \leq P_i \leq 2000.00$
- La suma de los N de un caso de prueba no será superior a 2097152.
- Tiempo: 1.5 segundos

● G

Julio: Torneos de Ajedrez - Reto Accenture

Los meses de vacaciones son los que tienen una mayor cantidad de torneos según la federación internacional de ajedrez. En los torneos de ajedrez clásicos obligan a apuntar las partidas. Esto permite justificar quién ganó la partida y hace que sea mucho más sencillo para los árbitros determinar los resultados.

Algunas planillas (documento en el que se apunta la partida y finalmente se entrega al árbitro) son imposibles de entender, a pesar de que las reglas de la federación dicen que la notación de estas debe ser legible.

Para poder entender estas planillas vamos a explicar las reglas de esta anotación:



1. Si se realiza el movimiento de una pieza mayor (todo menos peones) se anota con la inicial de la pieza D (Dama), R (Rey), T (Torre), A (Alfil), C (Caballo) y a continuación la casilla de destino. Por ejemplo: Da8, Ra2. (En caso de poder ir más de una pieza a una casilla, se pondría el lugar de la pieza de origen junto al destino, Dea8)
2. Si se realiza un movimiento de un peón, se pone el lugar a donde va el peón, por ejemplo d4.
3. Si se realiza una captura se añaden una x o un * entre la posición del peón o la inicial de la pieza mayor. Por ejemplo: D*a8, e*d4
4. Si se da un jaque, se denota con el símbolo + al final. Por ejemplo: D*a8+, e*d4+.
5. Si se realiza enroque corto, se denota como 0-0 y un enroque largo con 0-0-0 (enroque se refiere al único movimiento donde el rey puede mover dos veces intercambiándose con la torre)
6. Si se corona una pieza (en ajedrez, cuando un peón llega al final del tablero se puede convertir en cualquier pieza mayor). Se marcaría como e8=P siendo la P la letra inicial de la pieza mayor. Por ejemplo e8=D (un peón convertido a dama).
7. Si se da jaque mate se escribe ++ o # al final. Por ejemplo D*a8# o D*a8++.

Algunos ejemplos de anotación muy largos podrían ser d*e8=D++. (Existen dos peones que pueden comer en e8 en este caso, el de d come en e8 y, al llegar al final, se convierte en dama y da jaque mate) Todas las partidas que acaban en victoria son jaque mates, no se permite el abandono.

Entrada

La entrada consistirá en casos únicos. Primero aparecerá un número N denotando el número de jugadas de la partida. Seguidamente, vendrán N jugadas de la anotación. Todas las partidas comienzan con la posición inicial de las piezas y siguen una anotación correcta (en todas las piezas mayores la letra es mayúscula).

Salida

Debido a que en las partidas muchas veces el resultado no es legible, se pide hacer un programa que, analizando toda la planilla (es decir, la anotación de la partida), realice los diferentes movimientos necesarios para determinar el resultado final de la partida y facilitar la tarea al organizador de saber el resultado. Por cada caso de prueba se deberá decir, "1/2-1/2" en caso de empate, "1-0" en caso de victoria de blancas y "0-1" en caso de que negras ganen la partida.

Entrada de ejemplo

```
5  
e4 e5 Dh5 Re7 D*e5#
```

Salida de ejemplo

```
1-0
```

NOTA: El ejemplo de la entrada es conocido como el mate más corto de ajedrez, llamado el mate del loco.

Límites

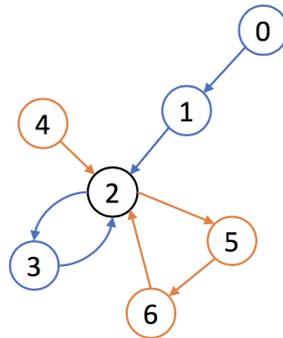
- $1 \leq N \leq 200$
- Tiempo: 1 segundo



Agosto: HH y el Bosón

Tal vez no habéis oído hablar sobre HH aún, pero es un gran científico y cree haber descubierto cómo hacer que varios bosones de Higgs choquen usando una red de tubos al vacío por donde los bosones se moverán.

Cada bosón comienza su camino por una serie de k_i tubos hasta un punto p en donde seguirá por otros c_i tubos que forman un ciclo (que comienza y termina en p). El punto p es el único punto en común entre todos los bosones y es donde ocurrirá la colisión de los bosones. HH tiene la teoría de que todos los c_i deben ser números coprimos y se ha asegurado de que sea así. Todos los tubos tienen exactamente la misma longitud y los bosones van exactamente a la misma velocidad. Si cada tubo se recorre en una unidad de tiempo, HH necesita saber en cuantas unidades de tiempo chocarán los bosones.



En la figura hay dos bosones (azul y naranja). El bosón azul tiene la trayectoria $0,1,2,3,2,3,2, \dots$ y el bosón rojo sigue la trayectoria $4,2,5,6,2,5,6, \dots$. Es obvio que después de 4 unidades de tiempo los bosones estarán ambos en el punto p (En este caso $p = 2$) y habrá una colisión.

Entrada

La entrada contiene varios casos de prueba. Cada caso de prueba comienza con una línea conteniendo un entero N (número de bosones). La entrada termina con un caso $N = 0$.

Las siguientes N líneas contienen la descripción del camino que seguirá cada bosón. Esta línea comienza con el número K de tubos en el camino seguido por los índices de los puntos extremos del tubo (T_i). Recuerda que el camino termina en un ciclo, y por lo tanto todos los caminos terminan en el punto P . Para ahorrar espacio, este punto P no está en la entrada.

Salida

Por cada caso de entrada debe haber una línea en la salida con el número de unidades de tiempo T que tardarán los bosones en colisionar en el punto p , módulo 100000007.

Entrada de ejemplo

```
2
3 0 1 2
4 3 1 4 5
2
3 0 1 2
5 3 4 1 6 5
3
8 0 1 2 3 4 5 6 7
6 8 2 9 10 11 12
10 13 14 15 2 16 17 18 19 20 21
0
```

Salida de ejemplo

```
1
5
206
```

Límites

- $2 \leq N \leq 8$
- $1 \leq K_i \leq 100$
- $1 \leq C_i \leq 30$
- $1 \leq T_i \leq 1000$
- Tiempo: 1 segundos



Septiembre: Vuelta a la rutina

Juan va a comenzar su segundo año de universidad, y con él llega una nueva oleada de sufrimiento: que si nuevas asignaturas y profesores, que si trabajos, que si papeleo, un montón de cosas pendientes por hacer, ... pfff. Solo de pensarlo se pone malo.

Para conseguir que la vuelta a la rutina sea lo menos dolorosa posible, se le ha ocurrido hacer una aplicación que le ayude a llevar la cuenta de todas las tareas que tiene pendientes. De esta forma, nunca se dejará nada olvidado, y le ayudará a evitar repetir la misma tarea múltiples veces, empezando así el curso con buen pie. ¿Podrías ayudarle?



Entrada

En la primera línea encontrarás dos números, N y M , separados por un espacio, que indican, respectivamente, el número de tareas diferentes que Juan tiene que hacer y el número de tareas que Juan ha realizado ya.

En cada una de las siguientes N líneas, encontrarás el nombre de cada tarea que Juan debe realizar. El nombre de la tarea cumple “[a-zA-Z]1,32” (sólo caracteres a-z, A-Z y espacio, entre 1 y 32 caracteres). Se garantiza que todas las tareas N_i a realizar son únicas.

En cada una de las siguientes M líneas, encontrarás el nombre de cada tarea que Juan ha realizado. Como Juan es un poco despistado, es posible que haga la misma tarea varias veces. Se garantiza que todas las tareas que Juan ha realizado aparecen en la lista de tareas a realizar.

Salida

Para cada caso de prueba, deberás decir “PONTE A TRABAJAR”, si tras consumir toda la entrada tiene tareas pendientes; “PUEDES DESCANSAR”, si todas las tareas han sido realizadas una sola vez; o “TRABAJAS DEMASIADO”, si todas las tareas han sido realizadas al menos una vez, pero alguna tarea ha sido hecha múltiples veces.

Entrada de ejemplo

```
1 2
Estudiar
Estudiar
Estudiar
```

Salida de ejemplo

```
TRABAJAS DEMASIADO
```

Entrada de ejemplo

```
4 4
Estudiar
Lavar los platos
Recoger habitacion
Hacer la compra
Hacer la compra
Lavar los platos
Hacer la compra
Recoger habitacion
```

Salida de ejemplo

```
PONTE A TRABAJAR
```

Límites

- $1 \leq N, M \leq 1000000$
- $1 \leq \text{Letras de tarea} \leq 32$
- Tiempo: 1,8 segundos



Octubre: Recogiendo caramelos

Maxx está muy entusiasmado por la gran fiesta de Halloween. Este día es caracterizado porque todos van disfrazados de un personaje (normalmente ambientado en el terror, la magia y lo oscuro) y salen de fiesta y festejan la noche de brujas. En ella, los niños suelen ir acompañados de otros niños y piden caramelos de puerta en puerta en cada casa.

Maxx está con sus inseparables amigos y esa noche la planean celebrar en grande, quieren recoger la mayor cantidad de caramelos posible en el menor tiempo que se pueda, ya que las calles de Madrid para la noche de brujas pueden ser un tanto inseguras para los niños a partir de cierta hora. Pero es tanto su afán, que ya se preparan desde Febrero para la gran fecha.

El plan del pequeño Maxx es sencillo, quieren ir de casa en casa (de forma siempre creciente) cogiendo dulces y, si ya no pueden guardar más dulces, planean ir rápidamente a casa de Maxx o de alguno de sus amigos, dejar los dulces y, rápidamente, continuar su búsqueda, es importante acotar que si un vecino da ligeramente más dulces de los que pueden cargar, los niños encontrarán la forma de llevarlos hasta la casa de alguno de sus amigos. Una vez allí, llenarán la bolsa vacía con los dulces que ya tienen en los bolsillos, pero teniendo la bolsa llena, no seguirán recogiendo más dulces. ¡Ten en cuenta que Maxx siempre irá a las casas con un índice mayor de la que salió y podrá ir a cualquiera que tenga un amigo siempre y cuando necesite descargar su bolsa de caramelos!

El tiempo que tardan Maxx y sus amigos en recoger caramelos es despreciable, ya que son increíblemente ágiles. Sin embargo, el tiempo que les toma ir de una casa A a otra casa B está dado por la distancia Manhattan de las 2 casas. Representando la ubicación de una casa en dos dimensiones, la distancia manhattan entre A y B sería $Manhattan(A, B) = |(A_x - B_x) + (A_y - B_y)|$

Maxx y sus amigos ya han investigado y saben cuántos caramelos les dará qué vecino, también saben los puntos X e Y de todas las casas y la capacidad K de la bolsa que van a llevar. Por último, saben que después de que pasen T unidades de tiempo, la noche genial de Halloween se podría convertir en un castigo para el mes de Noviembre. Por lo tanto, tienen que evitar a toda costa demorarse mucho en recoger la mayor cantidad de caramelos.

Maxx está interesado en ejecutar varias pruebas para ver de qué forma puede maximizar su ganancia en caramelos. ¿Podremos ayudarlo?

Entrada

La entrada está compuesta por múltiples casos de prueba.

Cada caso de prueba iniciará con 4 números; N , M , K y T , estos denotan, respectivamente, el número N de casas, el índice de las casas de sus M amigos (si $M_i = 3$, significa que el i -ésimo amigo vive en la tercera casa leída en la entrada), el tamaño de la bolsa K y el tiempo T que deben emplear como máximo en salir y volver de la casa de Maxx, se asume que esta casa siempre será la primera y que ningún amigo vive en casa de Maxx, es decir, no hay $M_i = 1$, esta casa es la única que no aporta caramelos. Después de eso, vendrán N líneas denotando la ubicación de cada una de las casas mediante puntos en un espacio bidimensional X_i e Y_i y la cantidad de caramelos N_i que esa casa puede dar a los niños separados por un espacio, finalmente, M líneas denotando que el amigo i vive en la casa M_i .

Salida

Para cada caso de prueba se debe imprimir un número, el número de caramelos obtenidos en la noche de Halloween.

Entrada de ejemplo

```
4 0 10 20
0 0 0
-1 -1 10
1 1 12
4 4 13
```

Salida de ejemplo

```
25
```

NOTA: En este ejemplo, lo mejor que puede hacer Maxx es ir a la tercera casa (2), coger 12 caramelos, volver a su casa (2), ir a la casa más lejana (8) y volver (8), pudiendo recoger 25 caramelos en 20 unidades de tiempo.

Límites

- $2 \leq N \leq 80$
- $0 \leq M < N$
- $0 \leq K \leq 80$
- $1 \leq T \leq 256$
- $1 \leq N_i \leq K$
- $1 \leq M_i, 0 \leq i \leq N$
- $-20 \leq X_i, Y_i \leq 20$
- Tiempo: 0.7 segundos



Noviembre: La casa encantada

Alice, Bob y sus amigos se lo han pasado muy bien en Halloween este año y han conseguido recoger una gran cantidad de caramelos. Todo esto gracias a sus conocimientos algorítmicos, que les han permitido decidir la ruta con la mayor cantidad de recompensas.

Pero ahora ha llegado el Día de Todos los Santos y los familiares de los niños tienen planeado visitar los cementerios con ramos de flores y acudir a misa. No es necesario decir que a los más jóvenes esto les parece muy aburrido. Aún emocionados con los eventos de la noche anterior, han conseguido escaquearse de ir a la misa con sus padres y han encontrado un escape room ambientado en una casa encantada.

Una vez todos han llegado al lugar, el organizador les cuenta las reglas y empieza el juego. El grupo se divide en dos equipos que empiezan en lugares separados de la casa encantada. El objetivo del juego es volver a encontrarse dentro de la mansión. Las habitaciones están numeradas desde el 1 hasta N y conectadas formando una cadena, una detrás de otra, en orden ascendente; de esta manera cada habitación H está conectada con $H - 1$ y $H + 1$, excepto los extremos que solo tienen una puerta. Algunas de ellas están bloqueadas y solamente pueden abrirse desde otro cuarto.

El equipo de Alice empieza en la habitación 1, mientras que el de Bob en la N , y avanzan hacia el otro equipo. Solamente se puede entrar en una habitación cerrada si uno de los equipos ha pasado por el cuarto desde el que se desbloquea. La última cosa a tener en cuenta es que solamente un equipo puede moverse en un instante dado, bloqueando los movimientos del otro. Hay que ser muy cuidadoso porque Alice aprovecha cualquier oportunidad para avanzar conforme se abra su puerta y muchas veces obligará al equipo de Bob a quedarse quieto.

Entrada

La primera línea indicará el número de casos de prueba T . Cada caso de prueba empezará con una línea con dos números N , indicando la cantidad de habitaciones, y C , indicando la cantidad de habitaciones cerradas. Las C líneas siguientes contendrán dos números cada una, H , indicando la habitación que está cerrada, y A , indicando la habitación desde la que se puede abrir H . Tanto H como A solamente aparecerán en el listado de habitaciones una vez por caso, ya sea como habitación cerrada o como habitación con mecanismo de apertura.

Salida

Por cada caso de prueba imprimir un único número indicando en qué habitación se encontrarán los dos equipos. En caso de no poder encontrarse se debe imprimir, "Encerrados para siempre."

Entrada de ejemplo

```
2
20 3
10 5
2 9
7 4
20 3
9 13
4 10
8 2
```

Salida de ejemplo

```
Encerrados para siempre.
10
```

Límites

- $1 \leq T \leq 10$
- $2 \leq N \leq 1000000$
- $1 \leq C \leq \text{Min}((N/2) - 1, 100000)$
- $2 \leq H_i, A_i \leq N-1$
- Tiempo: 0.5 segundos



Diciembre: Dora la Aisladora

Durante el mes de diciembre los virus proliferan en el ambiente y es mucho más fácil enfermarse. Para evitar otro caso similar al coronavirus, las autoridades han encargado a Dora la Aisladora que se encargue de proteger a la población para minimizar el riesgo de infección. La población se divide en regiones, pero solo existen carreteras entre ciertas regiones. Al aislar una región, todas sus carreteras quedarán incomunicadas y no será posible viajar a través de ella. La misión de Dora la Aisladora es encontrar qué región debe aislar para dividir a la población en el máximo número de grupos de regiones posible.



Entrada

La primera línea contendrá dos enteros N y M que representan el número de regiones y el número de carreteras entre ellas, respectivamente. Las siguientes M líneas contendrán dos enteros r_1 y r_2 que indican que existe una carretera entre las regiones r_1 y r_2 .

Salida

La salida se compone de un entero r que indica cuál es la región que aísla la población en el máximo número de grupos de regiones. En caso de que el máximo se alcance en varias regiones, Dora acudirá únicamente a la que tenga un menor identificador.

Entrada de ejemplo

```
10 10
0 4
1 7
2 3
3 4
3 7
4 7
5 9
6 7
6 9
8 9
```

Salida de ejemplo

```
7
```

Límites

- $1 \leq N \leq 200000$
- $1 \leq M \leq 200000$
- $0 \leq r_1, r_2 \leq N$
- Tiempo: 0.5 segundos