Preparación

ADA BYRON

MADRID

INTERNATIONAL GRANDMASTER

Competitive Programming

UPV

Problema número 250

El desgaste de los bombines

Tiempo máximo: 2,000-3,000 s Memoria máxima: 8192 KiB

http://www.aceptaelreto.com/problem/statement.php?id=250

Todas las urbanizaciones cerradas con un gran número de vecinos experimentan el mismo problema con las cerraduras: al ser tanta gente entrando y saliendo diariamente, las cerraduras se resienten y terminan funcionando mal.

Cuando en una puerta concreta se utiliza más uno de los dos lados de la cerradura, una solución que aplica habitualmente el personal de mantenimiento es *dar la vuelta al bombín* para que el desgaste de ambos lados se equilibre y, al menos, ambos lados funcionen más o menos igual.

Lamentablemente con la crisis el personal de mantenimiento de nuestra urbanización se ha tenido que eliminar por completo y sólo tenemos presupuesto para llamar al cerrajero *una vez*. Tenemos una previsión del número de veces que entrará y saldrá la gente por la puerta a lo largo de un periodo de tiempo y tenemos que decidir qué día queremos que venga el cerrajero a dar la vuelta al bombín para que al final del periodo el desgaste de ambos lados sea lo más parecido posible.

Entrada

La entrada estará compuesta por distintos casos de prueba que representan la estimación de uso de una puerta en un periodo de tiempo.

Cada caso de prueba consiste en dos líneas. La primera indica el número de días del periodo ($1 \leq n \leq 1.000.000$). La segunda línea contiene n números, cada uno de ellos indicando el desgaste del bombín; un número positivo v debe interpretarse como que el lado de fuera de la puerta se ha utilizado v veces más que el de dentro durante ese día. Un número negativo indica lo contrario. Se garantiza que el valor absoluto de v no superará 1.000.000 y que la suma de los valores absolutos de todos los números será menor o igual a 10^9.

La entrada termina con una línea con un 0.

Salida

Por cada caso de prueba se indicará al final de qué día querremos que venga el cerrajero para que dé la vuelta al bombín, de forma que al final del periodo de n días el desgaste de ambos lados de la cerradura sea lo más parecido posible. El primer día de la secuencia se corresponde con el número 1.

Si no es necesario que venga ningún cerrajero, se escribirá un 0. Si hay varios días posibles, preferiremos que venga cuanto antes, por si sube sus honorarios.

Entrada de ejemplo

```
4
1 2 2 1
4
1 2 2 0
4
1 2 -2 -1
4
6 1 -1 5
3
-4 -5 5
0
```

Salida de ejemplo

```
2
2
0
1
0
```

Autores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

Revisor: Alberto Verdejo.

Problema número 505

Volando drones

Tiempo máximo: 1,000-4,000 s Memoria máxima: 16384 KiB

http://www.aceptaelreto.com/problem/statement.php?id=505

Desde hace algún tiempo, quedamos la pandilla para ir a volar drones los sábados. Cada dron necesita dos pilas para poder volar: una de 9 V y otra de 1.5 V. En el club tenemos dos cajas para guardar las pilas, en una tenemos las de 9 V y en otra las de 1.5 V. Cada sábado cogemos las que están más cargadas y las colocamos en los drones. Para aprovechar al máximo el tiempo de vuelo, colocamos siempre las dos pilas más cargadas de cada tipo juntas, ya que los drones solo vuelan mientras las dos pilas tienen carga; después las dos siguientes más cargadas las ponemos en el siguiente dron; y así mientras queden pilas con carga de los dos tipos. Una vez colocadas las pilas, echamos los drones a volar. Cuando todos ellos acaban en el suelo por agotamiento de alguna de sus pilas, volvemos al club y guardamos en las cajas las pilas que todavía no están totalmente gastadas.

Por ejemplo, si a un dron le pusimos una pila de 9 V que permitía volar 5 horas y una pila de 1.5 V que permitía volar 2, el dron habrá volado 2 horas y al volver al club guardaremos la pila de 9 V a la que le quedarán 3 horas de vuelo. La pila de 1.5 V estará agotada y la echaremos al cubo de reciclaje.

Queremos saber cuántas horas de vuelo realizarán entre todos los drones cada sábado que podamos salir a volar, antes de que se agoten las pilas que hay ahora mismo en las cajas. Las pilas las tenemos que colocar en el club, por lo que cada dron solo puede volar una vez cada sábado.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con una línea en que se indica el número N de drones que tenemos ($1 \leq N \leq 1.000$), el número A de pilas de 9 V y el número B de pilas de 1.5 V ($1 \leq A,B \leq 200.000$). En la línea siguiente se indica el número de horas de vuelo que permite la carga de cada una de las pilas de 9 V y en la tercera línea el número de horas de vuelo que permite cada una de las pilas de 1.5 V.

Salida

Para cada caso de prueba se escribirá en una línea el número de horas de vuelo que realizarán los drones cada sábado, mientras se pueda salir a volar algún dron.

Se garantiza que esos números nunca serán mayores que 10^9.

Entrada de ejemplo

```
2 4 2
5 12 7 15
10 10
2 4 2
5 12 7 15
20 20
3 3 3
25 15 10
20 20 5
1 4 6
5 9 2 6
7 3 3 1 6 4
```

Salida de ejemplo

```
20
27 12
40 5
7 6 4 2 2 1
```

Autores: Isabel Pita y Alberto Verdejo.

Revisores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

Problema número 445

Semana de la Informática

Tiempo máximo: 1,000-2,000 s Memoria máxima: 32768 KiB

http://www.aceptaelreto.com/problem/statement.php?id=445

En breve comenzará la *Semana de la Informática*. La facultad ha organizado multitud de actividades (conferencias, seminarios, talleres, jornadas de puertas abiertas). Yo he estado mirando la planificación y he ido apuntando todas las actividades que me parecen interesantes. El problema es que son tantas que algunas solapan en el tiempo, por lo que no podré asistir a todas. Lo que he pensado es convencer a algunos compañeros para que asistan a las actividades a las que no puedo ir yo para que entre todos asistamos a todas las actividades que me han parecido interesantes, y después compartamos lo aprendido.

Dada la lista de actividades, de las que conozco cuándo empiezan y terminan, ¿podrías ayudarme a calcular cuántos compañeros necesito como mínimo para cubrir todas las actividades? Puedes tener en cuenta que aunque ni mis compañeros ni yo tenemos el don de la ubicuidad, sí tenemos el poder de teletransportarnos, por lo que en cuanto termina una actividad podemos *saltar* a donde comience otra, sin perdernos nada.

Entrada

La entrada consta de una serie de casos de prueba. Cada uno comienza con una línea con el número N de actividades interesantes ($1 \leq N \leq 200.000$). A continuación aparecen N líneas, cada una con dos números que representan el momento de comienzo y de finalización de una actividad (el comienzo siempre es estrictamente menor que la finalización). Estos tiempos son números enteros entre 0 y 10^9.

Salida

Para cada caso de prueba se escribirá una línea con el mínimo número de compañeros que harán posible que, junto conmigo, podamos asistir a todas las actividades, y de tal forma que ninguno tenga que estar a la vez asistiendo a dos de ellas.

Entrada de ejemplo

```
3
1 5
3 10
6 12
2
5 10
1 5
3
1 5
2 6
3 7
```

Salida de ejemplo

```
1
0
2
```

Autor: Alberto Verdejo.

Revisores: Marco Antonio Gómez Martín, Isabel Pita y Pedro Pablo Gómez Martín.

Problema número 396

Las partituras de la orquesta

Tiempo máximo: 2,000 s Memoria máxima: 8192 KiB

http://www.aceptaelreto.com/problem/statement.php?id=396

Para una orquesta modesta, gran parte del presupuesto se va en la compra de las partituras. Si cada músico tiene su propia copia, el número puede ascender a más de 100.

Afortunadamente, los músicos que tocan el mismo instrumento se sientan juntos y pueden compartir atril. Eso supone un ahorro considerable, aunque si se abusa demasiado puede provocar un efecto feo en la sala de conciertos. Al fin y al cabo ver a muchos músicos agolpándose alrededor del mismo atril no es muy estético.

La sección de asuntos económicos de la orquesta nos ha informado del número de partituras que podremos comprar para la próxima obra que se tocará. Teniendo en cuenta el número de músicos de cada instrumento, ¿cuál será el atril más concurrido?

Por ejemplo, si tenemos 8 violines, 5 violas, 5 violonchelos y 2 contrabajos y hay presupuesto para 6 partituras, podemos comprar dos copias de violines, dos de violas, una de violonchelos y otra de contrabajos. En ese caso el atril más compartido será el de los violonchelos, con 5 músicos.

Entrada

La entrada está compuesta de distintos casos de prueba, cada uno formado por dos líneas. La primera contiene dos enteros, p y n, con el número de partituras que podremos comprar (hasta 200.000) y el número de instrumentos *distintos* que hay en la orquesta (hasta 100.000). Se garantiza que se podrá comprar al menos una partitura para cada tipo de instrumento.

La segunda línea contiene n números positivos que indican la cantidad de músicos que hay en la orquesta para cada uno de los instrumentos (como mucho 1.000).

Salida

Por cada caso de prueba aparecerá un único entero indicando el número de músicos que se agolparán en el atril más compartido cumpliendo las restricciones de la orquesta y presupuesto. Recuerda que el objetivo es minimizar la cantidad de gente en el atril más concurrido.

Entrada de ejemplo

```
4 4
8 5 5 2
6 4
8 5 5 2
7 4
8 5 5 2
```

Salida de ejemplo

```
8
5
4
```

Autor: Marco Antonio Gómez Martín.

Revisores: Pedro Pablo Gómez Martín y Alberto Verdejo.

Problema número 386

Clúster de microondas

Tiempo máximo: 1,000-2,000 s Memoria máxima: 20480 KiB

http://www.aceptaelreto.com/problem/statement.php?id=386

Ante los retrasos continuos en la hora de llegada de los alumnos, los profesores que dan clase a primera hora de la tarde de cierta Facultad han decidido tomar medidas y no dejar entrar a nadie que llegue más de cinco minutos tarde. ¡Se acabaron las interrupciones!

Desde Delegación de Alumnos han contraatacado, culpando al Decanato de la situación. A medio día, argumentan, se forman unas grandes colas en los microondas disponibles para uso libre, lo que hace que haya que esperar mucho tiempo hasta conseguir tener la comida caliente. Y con lo aburridas que son algunas clases, aseguran, no todas las asignaturas se merecen renunciar por ellas a comer los macarrones calientes.

Antes de que llegue la sangre al río ante semejantes comentarios, desde Decanato han pensado que igual habría que hacer una ampliación de *hardware* en el clúster de microondas de la Facultad para evitar males mayores. Desde Asuntos Económicos miran con preocupación la medida, de modo que han pedido que se haga un estudio del número de microondas que se necesitaría tener disponibles para mantener la espera de sus usuarios dentro de unos márgenes razonables.

Y el Decanato ha querido involucrar a Delegación en él, para evitar críticas futuras... y les ha pedido que hagan ellos la estimación.

Entrada

El programa deberá leer de la entrada estándar múltiples casos de prueba.

Cada uno comienza con dos números enteros, n y t, indicando respectivamente cuánta gente utiliza los microondas a lo largo de un día cualquiera (como mucho 50.000), y cuál es el tiempo de espera máximo permitido.

Después aparecerá una línea con n parejas de números, una por cada hambriento usuario. El primer número de cada pareja indica el instante de tiempo en el que llega a hacer uso de un microondas, y el segundo el tiempo total (mayor que 0) que tiene que utilizarlo para calentar completamente su comida. Los usuarios estarán ordenados por orden de llegada. Hay veces que llegan en grupo, por lo que tendrán el mismo instante de llegada; en ese caso usarán los microondas en el orden en el que aparecen en la entrada.

La entrada terminará con un caso, que no deberá procesarse, en el que no hay nadie que quiera usar los microondas ($n = 0$).

Por simplicidad, los tiempos son números naturales carentes de unidad.

Salida

Por cada caso de prueba se escribirá el mínimo número de microondas que tiene que haber en el clúster para que nadie espere más tiempo del máximo permitido antes de empezar a calentar su comida. Se asume que hay "fila única", y en cuanto un microondas queda libre (sea cual sea), la siguiente persona en la cola comienza a utilizarlo sin dilación en un tiempo despreciable.

Entrada de ejemplo

```
2 5
0 5 0 3
3 5
0 6 0 3 10 4
0 0
```

Salida de ejemplo

```
1
2
```

Autor: Pedro Pablo Gómez Martín.

Revisores: Alberto Verdejo y Marco Antonio Gómez Martín.

Problema número 389

Aritmética verbal

Tiempo máximo: 2,000-3,000 s Memoria máxima: 4096 KiB

http://www.aceptaelreto.com/problem/statement.php?id=389

La cantidad de acertijos, pasatiempos y rompecabezas que han ido surgiendo a lo largo de los siglos es inmensa. Entre ellos, los *puzzles matemáticos* constituyen una categoría en sí misma, con acertijos sobre aritmética, combinatoria, topología o probabilidad.

Hoy nos quedaremos con la llamada *aritmética verbal*, o *criptaritmética*. En ella, los puzzles son operaciones matemáticas en las que los dígitos han sido sustituídos por letras, y hay que encontrar los números originales.

$$\begin{array}{r} \texttt{ada} \\ +\ \texttt{byron} \\ \hline \texttt{molar} \end{array} \quad \Rightarrow \quad \begin{array}{r} 767 \\ +\ 39508 \\ \hline 40275 \end{array}$$

Entrada

Cada caso de prueba es una operación aritmética (suma o producto) en la que los dígitos han sido sustituídos por letras minúsculas del alfabeto inglés. Se proporcionan los dos operandos separados por el operador ("`+`" o "`*`"), seguidos de un símbolo igual y el resultado de la operación, también en letras. Los números y los operadores están separados por un espacio.

Los *operandos* no tendrán más de 8 letras minúsculas; además, se garantiza que no habrá más de 10 letras diferentes en total.

Salida

Para cada caso de prueba el programa escribirá la operación asociada tras convertir las letras a dígitos, de modo que todas las apariciones de la misma letra se conviertan al mismo dígito y viceversa, y la operación aritmética sea correcta. Se garantiza que la solución será única.

Se debe añadir un espacio antes y después de cada operador. *No* se considera válida una asignación en la que cualquiera de los números tenga ceros *superfluos* a la izquierda, pero ten en cuenta que alguno de los valores podría ser `0`, que sí es válido.

Entrada de ejemplo

```
ada + byron = molar
ada * byron = leyenda
acepta + elreto = mental
solo + sola = baile
```

Salida de ejemplo

```
767 + 39508 = 40275
202 * 36951 = 7464102
473924 + 356321 = 830245
6797 + 6793 = 13590
```

Autor: Pedro Pablo Gómez Martín.

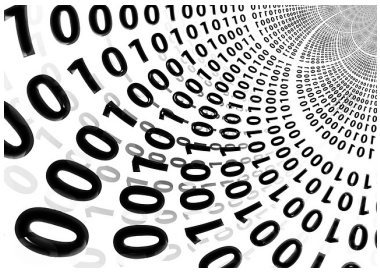
Revisor: Marco Antonio Gómez Martín.

Problema número 317

El código de la T.I.A.

Tiempo máximo: 1,000-4,000 s Memoria máxima: 4096 KiB

http://www.aceptaelreto.com/problem/statement.php?id=317



Ante la nueva oleada de ataques a las redes de otras agencias de investigación, la T.I.A. ha decidido tomarse en serio la transmisión de sus mensajes entre las distintas oficinas y ha contratado los servicios de un especialista que se hace llamar Pepe Gotera.

A la vista de la solución del supuesto experto, salta a la vista que vive anclado en el pasado. Lo único que ha hecho ha sido asignar a cada símbolo un número de entre uno y tres dígitos y ha instruido a los agentes especiales para que, cuando tengan que mandar un mensaje, simplemente sustituyan las letras y símbolos del mensaje por esos códigos, dando lugar a una secuencia de números, uno detrás de otro.

Por ejemplo, en una codificación donde la A es sustituida por el 12 y la B es sustituida por el 42, el grupo musical ABBA se codifica 12424212.

Pepe Gotera además de dar el método de encriptado ha proporcionado distintas tablas de codificación que los agentes irán utilizando a lo largo del tiempo. Sin embargo, nuestro "experto" no ha oído hablar de los códigos de Huffman[1] ni nada que se le parezca, por lo que cuando le ha llegado a Otilio el primer mensaje cifrado con una de esas tablas, se ha dado cuenta de que en realidad hay varios posibles descifrados distintos. El pánico se ha instalado en las oficinas centrales de la T.I.A. y cuando le han pedido explicaciones a Pepe Gotera ha dicho que, dado que ninguno de los códigos de las letras contiene ceros, se puede decir a los agentes que utilicen ceros para separar palabras.

El Super no está aún convencido de la efectividad de la solución, así que nos ha pedido ayuda. Dada la tabla de símbolos y un texto cifrado, ¿de cuántos mensajes distintos podría provenir?

Entrada

La entrada estará compuesta por distintos casos de prueba, terminados con una línea con un 0.

Cada caso de prueba está compuesto por tres líneas. La primera contiene el número N de símbolos que tiene la tabla de codificación. La segunda línea contiene N números correspondientes a los códigos de cada símbolo. Se garantiza que no hay números repetidos y que ninguno de ellos tiene ceros.

La tercera línea de cada caso de prueba tendrá el mensaje cifrado (una sucesión de entre 1 y 1.000 dígitos) codificado con la tabla anterior. En caso de tener ceros, éstos nunca estarán al principio ni al final del mensaje.

Salida

Para cada caso de prueba se escribirá en una línea independiente el número de mensajes que pueden dar el texto cifrado dado. Como puede haber muchos, se dará el resultado módulo 1.000.000.007.

Entrada de ejemplo

```
3
1 2 22
1221
3
1 2 22
12021
3
1 2 22
12321
0
```

[1] Los códigos de Huffman son *códigos prefijo*, es decir, que cumplen que la codificación de un símbolo nunca es prefijo de la codificación de ningún otro símbolo.

Salida de ejemplo

```
2
1
0
```

Autor: Marco Antonio Gómez Martín.

Revisores: Alberto Verdejo y Pedro Pablo Gómez Martín.

Problema número 706

El ingenioso Willy

Tiempo máximo: 1,000-3,000 s Memoria máxima: 24576 KiB

http://www.aceptaelreto.com/problem/statement.php?id=706

William Francis Sutton, también conocido como el ingenioso Willy, fue un famoso ladrón de bancos de Estados Unidos que cometió más de 100 atracos en las primeras décadas del siglo XX. La cantidad total de dinero robado se estima en unos 2 millones de dólares de la época.

Para la historia ha quedado como un maestro de los disfraces que conseguía embaucar a cualquiera de tal forma que nunca tuvo que disparar una sola bala. De hecho, él mismo aseguró en una entrevista al final de sus días que las armas que usaba nunca iban cargadas, pues alguien podría resultar herido.

Lo que pocos saben es que también era muy metódico planificando sus crímenes. Antes de entrar en un banco establecía perfectamente su plan de huida hasta su guarida, que alcanzaba siempre utilizando el camino más corto hasta ella. Arnold Schuster, un detective aficionado, conocía perfectamente su *modus operandi* así que cuando sospechaba que Willy iba a actuar ponía vigilancia en todas las calles por las que este podría pasar.

Pero Willy no era tonto. Si al entrar en un banco veía a Arnold merodeando sabía que sus rutas de escape no eran seguras y buscaba la siguiente alternativa más rápida.

Entrada

La entrada está formada por distintos casos de prueba cada uno representando una situación de atraco.

La primera línea de cada caso de prueba tiene dos números, NI (hasta 10.000) y NC (hasta 100.000) representando el número de intersecciones y el número de calles de la ciudad. A continuación aparecen NC líneas con la información de cada calle: dos números (entre 1 y NI) con las intereseciones que unen seguidos de un número que indica el tiempo que se tarda en recorrer esa calle en cualquier sentido (como mucho 10.000).

En cada caso de prueba el banco está situado en la intersección 1 y la guarida de Willy en la NI, y siempre hay al menos un camino desde el banco a la guarida.

Salida

Por cada caso de prueba se escribirá el tiempo que tardará Willy en llegar desde el banco a su guarida si esquiva las calles que podría haber utilizado con su plan original de seguir el camino más corto. Ten en cuenta que lo que debe evitar son las calles pero la ruta alternativa que siga puede pasar por interesecciones por las que también pasaba el plan original.

En caso de no poder llegar a la guarida se escribirá `IMPOSIBLE`.

Entrada de ejemplo

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

Salida de ejemplo

```
3
IMPOSIBLE
```

Autor: Marco Antonio Gómez Martín.

Revisores: Alberto Verdejo y Pedro Pablo Gómez Martín.

Problema número 691

Recaudación de la taquilla

Tiempo máximo: 2,000 s Memoria máxima: 16384 KiB

http://www.aceptaelreto.com/problem/statement.php?id=691

Esto del cine está fatal. Con tanta televisión por cable la gente cada vez viene menos, y la recaudación de la taquilla no llega ni para cubrir gastos. El gerente quiere que eso cambie y se le ha ocurrido la siguiente estrambótica idea: el coste de una entrada será igual al número de asientos libres en la fila seleccionada en el momento de la compra. Pretende embaucar a los clientes haciéndoles creer que así estarán más cómodos (a mí me da que no se lo van a creer...).

Lo que ocurre es que la gente que tiene abono anual tiene preferencia, entran antes al cine y se sientan como quieren, por lo que la ocupación de las filas cuando se empieza a vender entradas en taquilla no es uniforme.

Conociendo esta distribución inicial por filas, ¿nos ayudas a calcular cuánto es lo máximo que podemos recaudar hoy vendiendo entradas a la gente que está esperando para comprarlas? El taquillero es quien tiene la potestad de elegir dónde sienta a cada nuevo cliente y las entradas se venden siempre de una en una.

Entrada

En la entrada aparecen una serie de casos de prueba. Cada caso consta de dos líneas. En la primera aparecen tres números: el número F de filas que tiene el cine ($1 \leq F \leq 100.000$), el número A de asientos en cada fila (todas las filas son iguales, y $1 \leq A \leq 100.000$) y el número C de clientes en la cola esperando a comprar una entrada ($0 \leq C \leq 1.000.000$). En la siguiente línea aparecen F números (entre 0 y A), separados por espacios, indicando el número de asientos ocupados de cada fila, desde la más cercana a la pantalla a la más lejana, cuando comienza la venta de entradas.

Salida

Por cada caso de prueba se escribirá una línea que contenga la máxima recaudación que podemos obtener si seguimos la ocurrente estrategia propuesta por el gerente. Obviamente, si el cine se llena antes de que la cola de clientes se termine, la taquilla cerrará y esos clientes lamentablemente se quedarán sin entrada.

Entrada de ejemplo

```
4 4 2
4 2 1 3
1 5 6
0
```

Salida de ejemplo

```
5
15
```

Autor: Alberto Verdejo.

Revisores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

Problema número 507

Tándem

Tiempo máximo: 1,000-4,000 s Memoria máxima: 8192 KiB

http://www.aceptaelreto.com/problem/statement.php?id=507

La afición de nuestra familia a la bicicleta viene de muchos años atrás. Aún recuerdo las grandes tardes en las que tres generaciones de Cantadores nos lanzábamos al camino del Alto del Lirón mochilas en mano para merendar en la cumbre unos bocadillos de chorizo.

Ahora que no estoy en la generación de los más jóvenes sino en la intermedia, me toca perpetuar la tradición y recordar a los mayores que deben mantener la forma y a los jóvenes que el mundo de las actividades en familia es apasionante.

Como no siempre consigo mi objetivo, y con la esperanza de que la novedad me ayude, he comprado un *tándem*, una de esas bicicletas que se usan en pareja. El resultado ha sido un éxito; ahora todos están esperando la próxima excursión y preguntándose con quién les tocará montar. Y yo, como dueño del tándem, tengo que tener cuidado de que el peso de la pareja no supere la resistencia tope de mi flamante adquisición.

Entrada

La entrada estará compuesta por varios casos de prueba, cada uno ocupando dos líneas. La primera contiene dos números: el número de integrantes de la familia (al menos 1 y como mucho 200.000) y el peso máximo que el tándem puede aguantar (entre 0 y 2×10^9). La segunda línea contiene el peso de cada uno (números entre 0 y 10^9) separados por espacios.

La entrada termina con una línea con dos ceros, que no debe procesarse.

Salida

Por cada caso de prueba se escribirá una única línea que contiene el número de emparejamientos distintos que se pueden hacer sin romper el tándem.

A la hora de contar las parejas no se distinguirá quién se coloca delante o detrás, por lo que el emparejamiento en el que el abuelo va delante y el nieto detrás es el mismo que el que los coloca al revés.

Entrada de ejemplo

```
3 8
3 4 5
3 8
3 3 3
5 7
3 1 4 2 5
0 0
```

Salida de ejemplo

```
2
3
8
```

Autor: Marco Antonio Gómez Martín.

Revisores: Alberto Verdejo y Pedro Pablo Gómez Martín.

Problema número 385

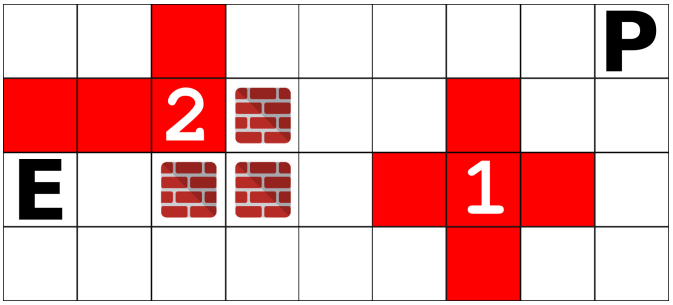
La ronda de la noche

Tiempo máximo: 1,000-2,000 s Memoria máxima: 16384 KiB

http://www.aceptaelreto.com/problem/statement.php?id=385

En Ciudad Decoro les preocupa que los jóvenes se reúnan con sus parejas en la noche, a espaldas de sus padres. Por ello en la última Ordenanza Municipal se ha obligado a que todas las casas tengan un jardín-laberinto para acceder a su puerta principal y que (opcionalmente) se instalen sensores de movimiento en los mismos. Conscientes de la posible dificultad de implantación de la Ordenanza, se ha decidido que todos los jardines-laberinto estén organizados en cuadrículas donde cada casilla está libre, tiene un muro o alberga un sensor de movimiento.

Como no podría ser de otra forma, la plataforma activista *#FreeLove* va a hacer todo lo posible para evitar que estas nuevas medidas supongan un obstáculo real para el amor. Durante las últimas semanas ha ido investigando y realizando mapas de todos los jardines-laberinto, anotando la posición de los sensores de movimiento. *#FreeLove* ha detectado que únicamente hay 10 tipos diferentes de sensores de movimiento, etiquetados como *CAT-k* donde k es un número natural entre 0 y 9. Los sensores captan movimiento en línea recta en las 4 direcciones (norte, sur, este y oeste) y la k de su categoría indica el número de casillas que cubren en cada dirección. De esta manera, un sensor *CAT-0* únicamente capta movimiento en la casilla en la que está instalado, mientras que un sensor *CAT-2* capta movimiento en su casilla y en 2 casillas en cada una de las 4 direcciones (en total cubre 9 casillas). Sin embargo, lo más interesante que ha descubierto *#FreeLove* es que los sensores no pueden traspasar los muros del laberinto, así que en ocasiones su alcance en alguna dirección se ve limitado.



#FreeLove necesita descubrir cuáles de los jardines-laberinto permiten que un amante vaya de la entrada del jardín a la puerta principal de la casa sin ser descubierto por ningún sensor. ¿Podrías ayudarles a diferenciar los jardines-laberinto impenetrables de aquellos favorables al amor, y en esos casos calcular el mínimo número de casillas que hay que atravesar para llegar de la entrada del jardín a la puerta principal?

Entrada

La entrada comienza con una línea conteniendo el número de jardínes a analizar. Cada jardín comienza con dos números $0 < ancho, alto \leq 1000$ con el ancho y alto del jardín-laberinto en una línea. Le sigue la descripción del jardín en *alto* líneas de *ancho* caracteres cada una. Estos caracteres son:

- '`#`': Una pared.
- '`.`': Una casilla libre.
- '`E`': La casilla donde está la entrada al jardín-laberinto.
- '`P`': La casilla donde está la puerta principal de la casa.
- k, con $0 \leq k \leq 9$: Casilla que alberga un sensor *CAT-k*.

La entrada al jardín y la puerta principal de la casa pueden estar en cualquier parte del jardín, incluidas casillas interiores. Además, nada impide que algún sensor vigile la casilla donde está la entrada al jardín o la puerta principal de la casa, por lo que en esos casos no será posible recorrer el jardín sin ser descubiertos.

Salida

Por cada jardín-laberinto la salida será una línea con el mínimo número de casillas del jardín que hay que atravesar para llegar de la entrada a la puerta principal de la casa sin ser descubierto, o la palabra `NO` en caso de que sea imposible.

Entrada de ejemplo

```
3
9 4
........P
..2#.....
E.##..1..
.........
6 4
....#P
..#...
.####.
E....1
10 5
E.........
..........
....2.....
..........
.........P
```

Salida de ejemplo

```
12
10
NO
```

Autor: Enrique Martín Martín.

Revisores: Alberto Verdejo y Pedro Pablo Gómez Martín.

Problema número 502

Hoy comemos mejillones

Tiempo máximo: 1,000-4,000 s Memoria máxima: 16384 KiB

http://www.aceptaelreto.com/problem/statement.php?id=502

Mis padres han comprado 5 kilos de mejillones. Nos gusta comerlos simplemente cocidos al vapor con unas gotas de limón. Para evitar que se llenen los platos con las conchas, se sirven ya abiertos en fuentes enormes. Aun así, si no tienes cuidado, pronto tu plato estará completamente lleno y no podrás colocar más mejillones para comerlos. Mi hermano ha ideado un sistema para mantener el plato lo más libre posible. Va colocando las conchas vacías con la parte interior hacia arriba, y las amontona poniendo siempre una concha más pequeña sobre una más grande, de forma que se coloca en el interior y la torre tiene mayor estabilidad.

Hoy he decidido hacer lo mismo para intentar tener más sitio en el plato y así comer más. Sin embargo, al recoger los platos me he dado cuenta de que el suyo tenía muchos menos montones que el mío, a pesar de que habíamos comido lo mismo. ¿Sabes cuál es la estrategia que utiliza mi hermano para minimizar el número de montones en el plato?

Debes tener en cuenta que los mejillones hay que comerlos en el orden en que te los van sirviendo, no vale elegirlos. Buscar el mejor de la fuente se considera de mala educación, y hay que conformarse con el que te corresponde en el orden en que se presentaron.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con una línea en que se indica el número N de mejillones que te ha tocado comer ($1 \leq N \leq 500.000$). A continuación aparece una línea con N enteros que representan el tamaño de las conchas de esos mejillones (números enteros entre 1 y 10^9).

Salida

Para cada caso de prueba se escribirá en una línea el menor número de montones de mejillones que se pueden formar con todas las conchas de los mejillones que te ha tocado comer.

Entrada de ejemplo

```
4
6 5 4 3
5
3 4 5 5 5
8
7 7 7 5 2 5 2 2
```

Salida de ejemplo

```
1
5
3
```

Autores: Isabel Pita y Alberto Verdejo.

Revisores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.