Preparación

ADA BYRON

MADRID

PUPIL

Competitive Programming
UPV

Problema número 572

Salvados por el 9

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

http://www.aceptaelreto.com/problem/statement.php?id=572

El profesor de Lengua de Mateo está revisando las notas de todos los alumnos para determinar la nota final de curso. Sigue un sistema de evaluación continua en el que registra una nota por cada una de las cinco actividades realizadas en clase. Sacar un 0 en una de las actividades significa que ha habido una ausencia total de participación en esa actividad por parte del alumno en cuestión. Solamente prestar la debida atención durante el desarrollo de la misma supone escapar del temido *rosco*. Según su sistema habitual ningún alumno que tenga un 0 en alguna de las actividades puede aprobar su asignatura, incluso aunque la media de sus notas supere el 5.

Sin embargo, ha decidido cambiar ligeramente su sistema porque se ha dado cuenta de que todo el mundo puede tener un mal día... si a cambio tiene otro día magnífico. Por eso ha decidido que no suspenderá directamente a los alumnos con algún 0 sino que calculará la media de sus notas en caso de que al menos tengan un 9 en alguna de las actividades.

Entrada

La entrada comienza con un número indicando la cantidad de casos de prueba que deberán procesarse. Cada caso de prueba se describe en una línea que contiene las cinco notas de un alumno (un número real entre 0 y 10 con exactamente un decimal).

Salida

Para cada caso de prueba, el programa escribirá `MEDIA` si según el nuevo sistema para ese alumno hay que calcular la media de sus notas, y `SUSPENSO DIRECTO` si queda directamente suspenso por tener 0 en alguna actividad y no haber sacado al menos un 9 en otra.

Entrada de ejemplo

```
3
5.0 7.5 4.0 6.5 0.8
4.8 0.0 6.3 4.5 9.5
8.5 7.0 8.2 0.0 1.2
```

Salida de ejemplo

```
MEDIA
MEDIA
SUSPENSO DIRECTO
```

Autores: Clara Segura y Alberto Verdejo.

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

Problema número 551

Mensaje secreto

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

http://www.aceptaelreto.com/problem/statement.php?id=551

Para enviarse mensajes secretos, Ana y Tomás (estudiantes de programación) escriben notas en papel en las que cambian cada letra o símbolo por su valor numérico (decimal) de acuerdo al código ASCII. Así, la letra `A` la escriben como `65`, la `B` como `66`, y así sucesivamente. Ana ha recibido el siguiente mensaje de Tomás:

`77101321031171151169711532109971153211311710132108111115321121171101161011141111546`

Y necesita ayuda para descifrarlo. ¿Puedes echarle una mano?

Entrada

La entrada está formada por un conjunto de mensajes codificados, cada uno en una línea distinta. Los mensajes pueden contener sólo letras mayúsculas y minúsculas del alfabeto inglés, además de los símbolos *espacio*, *coma* y *punto*. La longitud un mensaje, después de ser descifrado, no es nunca mayor de 100 caracteres.

Salida

Para cada mensaje codificado, se escribirá en una línea el mensaje descifrado.

Entrada de ejemplo

```
65666665
651009766121114111110321091111089746
77101321031171151169732112114111103114971099711432101110326746
```

Salida de ejemplo

```
ABBA
AdaByron mola.
Me gusta programar en C.
```

Autor: Luis Fernando Lago Fernández.

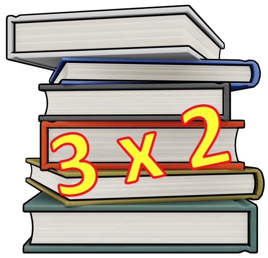
Revisores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

Problema número 650

Oferta 3×2

Tiempo máximo: 1,000-2,000 s Memoria máxima: 4096 KiB

http://www.aceptaelreto.com/problem/statement.php?id=650



En mi librería favorita, un día al año sacan una oferta de 3×2: por cada tres libros que compras te regalan uno, el de menor precio. Yo espero ansioso a que llegue ese día y entonces compro todos los libros que puedo.

El año pasado compré libros que valían 40, 35, 30, 25, 20, 15 y 10 euros, por los que pagué 150 euros, obteniendo un descuento total de 25 euros, ya que la librería utiliza un sistema estricto (¡y muy ventajoso para ella!) a la hora de seleccionar los libros que te regala: son siempre los de menor precio del lote.

Pero después me di cuenta de que si hubiera ido a comprar varias veces podría haber obtenido un descuento mayor. Por ejemplo, si primero hubiera comprado los libros con precios 40, 35, 30 y 25, me habrían regalado el de 25, y al comprar después el resto me hubieran regalado el que valía 10, obteniendo un descuento total de 35 euros.

¿Puedes ayudarme a decidir cómo debería comprar los libros que quiero este año para ahorrarme lo máximo posible?

Entrada

El programa deberá encontrar la solución a diferentes casos de prueba. Cada caso consta de dos líneas. En la primera aparece el número (entre 1 y 1.000) de libros que quiero comprar. En la segunda aparecen los precios de los libros (entre 1 y 10.000), separados por espacios.

Salida

Para cada lote de libros se escribirá el descuento máximo que puedo obtener si me aprovecho de repartirlos en varias compras.

Entrada de ejemplo

```
7
40 35 30 25 20 15 10
3
50 10 30
2
25 20
```

Salida de ejemplo

```
45
10
0
```

Autor: Alberto Verdejo.

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

Problema número 394

Hijos a tope

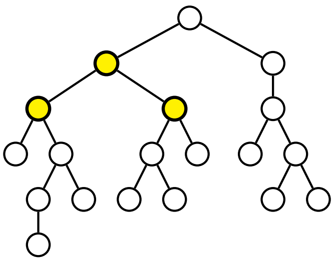
Tiempo máximo: 1,000-2,000 s   Memoria máxima: 4096 KiB

http://www.aceptaelreto.com/problem/statement.php?id=394

China es el país más poblado del mundo con más de un quinto de la población mundial. En 1953 empezó a realizarse en el país el primer censo moderno, revelando una población de cerca de 600 millones de habitantes. En los años 60 el gobierno comenzó a promover el retraso en la edad de contraer matrimonio y la planificación familiar. En 1972 se establecieron los primeros límites en el número de hijos por familia, siendo de dos en las ciudades. La política del hijo único se estableció en 1979 con objeto de frenar el crecimiento de la población, cuando esta ya estaba cercana a los 1000 millones de personas. Esta política combinaba el uso de la propaganda, la presión social, el establecimiento de beneficios e incluso penalizaciones económicas. Sin embargo, las minorías étnicas nunca han estado sujetas a dicha política, y en los entornos rurales a las parejas se les ha permitido tener otro hijo si su primogénito fue una niña. En 2015 se puso fin a la política del hijo único permitiendo de nuevo a todas las parejas tener dos hijos.

Un grupo de demógrafos chinos está haciendo un estudio con árboles genealógicos de familias chinas en las que se ha respetado el máximo de dos hijos que se estableció en 1972 y que recientemente se ha recuperado. En cada familia estudian las *subfamilias generacionalmente completas*, que son aquellas subfamilias en las que todos sus miembros han tenido exactamente dos hijos. Quieren saber cuál es la subfamilia generacionalmente completa con mayor número de generaciones.

Por ejemplo, en el siguiente árbol genealógico se ha resaltado la subfamilia generacionalmente completa con más generaciones, que en este caso son dos.



Entrada

La entrada comienza indicando el número de casos de prueba que vendrán a continuación. Cada caso consiste en una secuencia de números que pueden ser 0, 1 o 2 indicando el número de hijos de los miembros de la familia. Primero aparece el número de hijos de la raíz del árbol genealógico, y si tiene hijos, a continuación figuran las descripciones de las subfamilias de cada uno de sus hijos, que a su vez están descritas de la misma forma.

Los árboles genealógicos nunca contendrán más de 5.000 miembros.

Salida

Para cada árbol genealógico, se escribirá una línea con el número de generaciones de la subfamilia generacionalmente completa con mayor número de generaciones.

Entrada de ejemplo

```
5
0
2 0 0
1 2 1 0 1 0
2 2 0 1 2 0 0 2 0 0
2 2 2 0 2 1 0 0 2 2 0 0 0 1 2 0 2 0 0
```

Salida de ejemplo

```
0
1
1
2
2
```

Autores: Clara Segura y Alberto Verdejo.

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

Problema número 705

Saltando al otro lado

Tiempo máximo: 1,000-2,000 s   Memoria máxima: 8192 KiB

http://www.aceptaelreto.com/problem/statement.php?id=705

Juan Esal Tarín quiere saltar de un lado a otro de un río. La otra orilla parece lejos pero, afortunadamente, hay algunas piedras (conocidas también como *zamburguesas* en algunos contextos) a las que puede ir saltando hasta llegar a la orilla contraria.

Saltar de un sitio a otro, no obstante, es muy cansado. Cada vez que da un salto que le requiere usar su máxima potencia, esta se ve decrementada en una unidad. Eso significa que si en una orilla parte con una potencia de salto de K metros y las piedras están separadas distancias menores que ese K, podrá pasar sin problema al otro lado. Sin embargo, si una de las separaciones es de exactamente K, entonces las separaciones de las piedras siguientes no podrán exceder $K - 1$. Si algún salto requiere esa distancia, entonces su capacidad de salto bajará de nuevo, hasta $K - 2$.

Por ejemplo, en un río las piedras están colocadas a distancia 1, 6, 7 y 11 y la otra orilla a distancia 13. Si partimos con una capacidad de salto de 5, el proceso será el siguiente:

- Salta desde la orilla (posición 0) hasta la primera piedra, situada a distancia 1, lo que requiere una capacidad de salto 1.
- Desde la piedra en 1 salta a la piedra en 6. Como la distancia total coincide con la capacidad de salto, esta se ve reducida a 4.
- Desde la piedra en 6 salta a la piedra en 7 sin problema.
- Desde la piedra en 7 salta a la piedra en 11. Este salto también es posible pues requiere capacidad de salto 4, que coincide con la capacidad actual. Al ser iguales, la capacidad se ve reducida a 3.
- Desde la piedra en 11 salta, sin problema, a la orilla en 13.

Dado una colocación de las piedras en un río, ¿cuál es la mínima capacidad de salto de partida necesaria para poder atravesar el río sin problema?

Entrada

La entrada comienza con un número indicando el número de casos de prueba.

Cada caso de prueba empieza con un número N con el número de posiciones que se darán (hasta 100.000). Las primeras $N - 1$ posiciones se corresponden con las piedras en el río (dadas en orden de distancia a la orilla origen) y la última con la distancia a la otra orilla (como mucho 10^9).

Salida

Por cada caso se escribirá una línea con la capacidad de salto inicial mínima para llegar al otro lado.

Entrada de ejemplo

```
2
2
5 10
5
1 6 7 11 13
```

Salida de ejemplo

```
6
5
```

Autor: Marco Antonio Gómez Martín.

Revisores: Alberto Verdejo y Pedro Pablo Gómez Martín.

Problema número 646

Camiones de reparto

Tiempo máximo: 2,000-3,000 s Memoria máxima: 20480 KiB

http://www.aceptaelreto.com/problem/statement.php?id=646

Somos una empresa de transporte y hemos decidido renovar parte de nuestra flota de camiones de reparto. A nosotros nos conviene que los camiones sean anchos, porque así se puede repartir y colocar mejor la mercancia. Pero claro, hay ciudades con calles muy estrechas, por donde no todos los camiones pueden pasar.

Tenemos mapas actualizados de las ciudades donde trabajamos, donde hemos señalado para cada calle cuál es la anchura máxima que puede tener un camión para poder transitar por ella.

¿Nos ayudas a decidir si un camión de una anchura determinada puede circular por una ciudad para llegar desde un punto concreto a otro?

Entrada

La entrada está formada por una serie de casos de prueba. En cada caso, primero se describe una ciudad. La primera línea contiene el número V de intersecciones de la ciudad (numeradas de 1 a V) y el número E de calles entre intersecciones. A continuación aparecen E líneas, cada una con tres números: las intersecciones que une esa calle, y la anchura máxima que puede tener un camión que transite por ella. Todas las calles son de doble sentido.

Tras la descripción de la ciudad, aparece un número K de consultas, seguido de K líneas, cada una con tres números: dos intersecciones distintas, el origen y el destino, y la anchura de un camión, del que estamos interesados en saber si podría viajar desde el origen hasta el destino.

Todos los casos cumplen que $2 \leq V \leq 10.000$, $0 \leq E \leq 100.000$ y $1 \leq K \leq 10$. Todas las anchuras son números entre 1 y 1.000.000.

Salida

Para cada caso de prueba se escribirán K líneas, una por consulta. La respuesta a una consulta será `SI` si un camión de la anchura correspondiente podría recorrer un camino que le llevara del origen al destino, y `NO` en caso contrario.

Entrada de ejemplo

```
5 5
1 2 10
1 3 30
2 4 20
3 4 15
4 5 12
3
1 5 8
1 4 12
2 5 15
```

Salida de ejemplo

```
SI
SI
NO
```

Autor: Alberto Verdejo.

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

Problema número 313

Fin de mes

Tiempo máximo: 1,000-2,000 s Memoria máxima: 4096 KiB

http://www.aceptaelreto.com/problem/statement.php?id=313

A mí no me asusta el fin del mundo; me asusta el fin de mes, porque no siempre consigo que mis ingresos lleguen conmigo. Los gastos se acumulan, y no sé qué más hacer para estirar mi triste sueldo.

Creo que el primer paso para mejorar mi situación es hacer una estimación de lo bien o lo mal que me va a ir un mes, en función de los ingresos y los gastos previstos. Sé cuánto dinero tengo en el banco al principio, y sé cuánto va a variar. ¿Me ayudas a saber si llegaré a fin de mes con dinero en el banco?

Entrada

La entrada comienza con un número que indica cuántos casos de prueba vendrán a continuación. Cada caso contiene dos números, $-10.000 \leq s,c \leq 10.000$ indicando, respectivamente, el saldo en mi cuenta bancaria el primer día del mes, y el cambio estimado (ingresos menos gastos) durante el mes.

Salida

Para cada caso de prueba, se escribirá "`SI`" si llego a fin de mes con saldo mayor o igual que cero, y "`NO`" en otro caso.

Entrada de ejemplo

```
4
100 -10
-10 -100
-10 100
100 -1000
```

Salida de ejemplo

```
SI
NO
SI
NO
```

Autor: Pedro Pablo Gómez Martín.

Revisor: Marco Antonio Gómez Martín.

Problema número 503

Dados de rol

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

http://www.aceptaelreto.com/problem/statement.php?id=503

Desde el principio de los tiempos, los juegos de rol utilizan dados para tomar decisiones sobre cómo avanzar, cómo los hechizos afectan a los jugadores, elegir las condiciones climatológicas, etc. Los dados pueden tener diferentes formas, con diferente número de caras, de colores variados, con colores lisos o irisados, opacos o transparentes...

La forma en la que rueda el dado al ser lanzado y la equidad de los resultados dependen de la regularidad del poliedro que da forma al dado. De los seis dados que mayoritariamente se utilizan en los juegos de rol, cinco tienen la forma de los llamados *cinco sólidos platónicos*, los únicos cinco poliedros "perfectos" al tener caras regulares. Son el tetraedro regular (con 4 caras, triángulos equiláteros), el hexaedro regular o cubo (con 6 caras cuadradas), el octaedro regular (con 8 caras, triángulos equiláteros), el dodecaedro regular (con 12 caras, pentágonos regulares) y el icosaedro regular (con 20 caras, triángulos equiláteros). El sexto dado de los más utilizados tiene 10 caras y forma de trapezoedro pentagonal. También se utilizan dados simulados, por ejemplo de 2 o 3 caras, a partir de los anteriores mediante la división de números. Por ejemplo, el dado de 3 caras se puede obtener lanzando un dado de 6, dividiendo el resultado entre 2 y finalmente redondeándolo hacia arriba. Y después hay modificadores, que alteran el resultado de una tirada, con bonificaciones o penalizaciones. ¡Todo un mundo de posibilidades!

En el juego de rol al que ahora estamos jugando, el daño que inflige una daga al jugador se calcula de la siguiente manera: el jugador extrae dos dados de una bolsita, dice un número y lanza los dados; el daño será la diferencia (en valor absoluto) entre el número dicho y la suma de las caras de los dos dados.

Para tener ventaja en el juego quiero calcular cuáles son las sumas más probables, dependiendo de los dados que haya sacado. ¿Me ayudas?

Entrada

La entrada está formada por una serie de casos de prueba. El número de casos que vendrán a continuación aparece en la primera línea.

Cada caso consiste en dos números, las caras que tienen los dos dados extraídos. Los dados pueden tener entre 2 y 20 caras, que están numeradas desde 1 hasta el número de caras. En estos dados todas las caras son igualmente probables.

Salida

Para cada caso de prueba se escribirá una línea con el valor de la suma más probable. Si existieran varios valores igualmente probables, se escribirán todos ellos, ordenados de menor a mayor.

Entrada de ejemplo

```
2
6 6
4 6
```

Salida de ejemplo

```
7
5 6 7
```

Autor: Alberto Verdejo.

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

Problema número 495

Colocando campamentos de altura

Tiempo máximo: 1,000-2,000 s   Memoria máxima: 4096 KiB

http://www.aceptaelreto.com/problem/statement.php?id=495

Juanito quiere subir al Everest y está planeando la ascensión desde el campo base. Para llegar a la cumbre tendrá que instalar varios campamentos de altura en los que hacer noche.

Los campamentos sólo se pueden instalar en determinados puntos de la montaña, y la instalación de cada uno de ellos lleva un coste asociado que depende de su ubicación, el espacio disponible, la exposición al viento, etc. Juanito quiere minimizar los costes de la expedición, sabiendo que en cada jornada no puede superar más de un máximo conocido de desnivel desde el campamento de partida de ese día hasta el de llegada.

¿Puedes ayudarle a calcular el coste mínimo de la expedición?

Entrada

La entrada está formada por distintos casos de prueba, cada uno de ellos compuesto de 4 líneas.

La primera línea contiene un único número entero N ($5 \leq N \leq 1.000$), que representa el número de posibles ubicaciones para los campamentos de altura. La segunda línea contiene N números enteros, correspondientes a la altura de cada una de las ubicaciones, medida desde el campo base. Las ubicaciones están ordenadas por altura desde el campo base, y la última ubicación corresponde a la cima de la montaña. La tercera línea contiene N números enteros entre 10 y 100, correspondientes a los costes asociados a la instalación de un campamento en cada una de las ubicaciones. El último de estos números siempre será 0, pues la última ubicación es la cumbre y en la cumbre no se instala ningún campamento. La cuarta línea contiene un único número entero M ($100 \leq M \leq 1.000$), que representa el máximo desnivel permitido entre dos campamentos consecutivos (incluyendo el campo base), o entre el último campamento y la cumbre de la montaña.

Se garantiza que el problema siempre tiene solución.

Salida

Para cada caso de prueba, se escribirá una línea con el coste mínimo de la expedición.

Entrada de ejemplo

```
5
100 200 300 400 500
10 10 10 10 0
100
11
300 600 1000 1500 1700 2000 2150 2400 2700 2950 3300
10 20 30 40 50 60 70 80 90 100 0
650
```

Salida de ejemplo

```
40
250
```

Autor: Luis Fernando Lago Fernández.

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

Problema número 494

Asalto a la reprografía

Tiempo máximo: 1,000-2,000 s   Memoria máxima: 4096 KiB

http://www.aceptaelreto.com/problem/statement.php?id=494

Te apuntaste a un concurso de programación hace algún tiempo y, pese a tener buenos propósitos, al final has entrenado muy poco. Te has enterado de cuál es la reprografía que utilizan los organizadores para imprimir los cuadernillos con los problemas y la tentación ha sido demasiado fuerte. Estás decidido a entrar en ella furtivamente por la noche y hacerte con un ejemplar, para intentar partir con algo de ventaja.

Lamentablemente, la reprografía está custodiada por un guarda que hace la ronda por las inmediaciones. Bajo la luz de la luna, llevas desde medianoche apuntando los momentos en los que el vigilante ha pasado por delante de la puerta y estás seguro de que hay un ciclo, aunque de duración incierta, a lo largo del que el vigilante pasa quizá varias veces por la tienda.

Está empezando a amanecer y no puedes retrasar más tu asalto. El guarda acaba de pasar y con la información que tienes necesitas predecir en qué momento volverá a hacerlo, para saber si tendrás o no tiempo suficiente para entrar y salir con tu preciado objetivo.

Entrada

La entrada está formada por distintos casos de prueba, cada uno ocupando dos líneas.

En la primera línea aparece un único número $5 \leq N \leq 1.000$ con el número de veces que has visto pasar al guarda por delante de la puerta. La segunda tiene N números indicando la hora a la que ha pasado como el número de minutos transcurridos desde medianoche. Se garantiza que el primer número siempre es 0 y que el guarda tarda al menos 1 minuto y no más de 10 en volver.

El final de la entrada se marca con una línea con un único cero que no se debe procesar.

Salida

Para cada caso de prueba, se escribirá una línea con el minuto desde medianoche en el que se espera que el guarda vuelva a aparecer.

Se garantiza que en la entrada el ciclo se repite *completo* al menos dos veces. Ten en cuenta, eso sí, que el último ciclo puede no ser completo.

Entrada de ejemplo

```
10
0 1 2 4 5 6 8 9 10 12
6
0 2 4 6 8 10
8
0 1 3 6 7 9 12 13
0
```

Salida de ejemplo

```
13
12
15
```

Autores: Luis Fernando Lago Fernández y Pedro Pablo Gómez Martín.

Revisor: Marco Antonio Gómez Martín.

Problema número 501

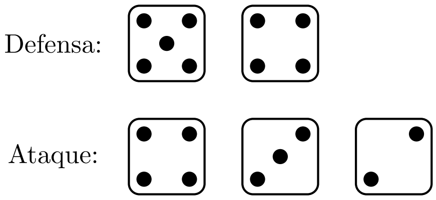
Pequeña batalla de dados

Tiempo máximo: 1,000-2,000 s   Memoria máxima: 4096 KiB

http://www.aceptaelreto.com/problem/statement.php?id=501

En el juego del Risk se conquistan territorios con una estrategia basada en dados. Las batallas empiezan cuando el atacante mueve un determinado número de tropas al territorio que desea conquistar. A continuación, la batalla se divide en oleadas. Para cada oleada, el atacante y el defensor lanzan un determinado número de dados. El número de dados que tira cada contrincante no puede ser mayor que el número de tropas disponibles, ni mayor que el número de dados disponibles. Tras tirar los dados cada bando ordena sus dados de mayor a menor y luego se alinean con los dados del enemigo. Finalmente, para cada par de dados defensor-atacante el atacante pierde una tropa si el valor de su dado es igual o menor que el valor del dado del defensor. En caso contrario, es el defensor quien pierde una tropa. Después de cada oleada, y si quedan tropas en ambos bandos, el atacante decide si retirarse o continuar con la batalla.

Veamos un ejemplo en el que el defensor tiene 2 tropas y el atacante tiene 3. El defensor tira dos dados y saca 4 y 5. El atacante tira tres dados y obtiene 3, 4 y 2. Tras ordenarlos y emparejarlos, los dados quedan como en la figura:



En este caso, en la primera y segunda pareja de dados el defensor tiene un valor mayor o igual que el atacante, por lo que este último pierde dos tropas. El tercer dado del atacante se descarta ya que no está emparejado con ninguno del defensor. Si el atacante continua para una segunda oleada, entonces el defensor lanza dos dados y el atacante uno. Si, por ejemplo, obtuvieran el defensor 1 y 5 y el atacante 6 entonces, dado que 6 es mayor que 5, el defensor perdería una tropa. En una tercera oleada cada bando lanza un único dado. Si ambos sacaran 5, entonces el defensor ganaría y la batalla terminaría con una tropa sobreviviente en el bando defensor y ninguna en el bando atacante.

Entrada

Cada batalla se describe en dos líneas. La primera línea tiene cinco números enteros con la información siguiente: número de tropas de defensa (td), número de tropas de ataque (ta), máximo número de dados a usar por la defensa (dd), máximo número de dados a usar por el ataque (da) y número de oleadas (no). Los límites para estos valores son: $1 \leq td \leq 100.000$, $1 \leq ta \leq 100.000$, $1 \leq dd \leq 1.000$ y $1 \leq da \leq 1.000$. La segunda línea incluye toda la secuencia de tiradas de dados en la batalla como: valores obtenidos por la defensa en la primera oleada, valores obtenidos por el ataque en la primera oleada (siempre se tira el máximo número de dados posible), y a continuación de forma equivalente para las sucesivas oleadas. Las oleadas están siempre completas, pero el atacante puede retirarse antes del final de la batalla sin que haya un ganador. Siempre hay al menos una oleada. El máximo número de lanzamientos de dados en una batalla es 200.000.

Salida

Para cada batalla, el programa debe imprimir una línea con el número de tropas que sobreviven para el defensor y el número de tropas que sobreviven para el atacante.

Entrada de ejemplo

```
2 3 3 3 3
4 5 3 4 2 1 5 6 5 5
10 10 5 5 2
1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
4 4 1 4 5
1 1 1 1 1 2 2 2 3 3 1 2 4 4 2 5 5 5 5 6 2
```

Salida de ejemplo

```
1 0
5 5
0 3
```

Autor: Gonzalo Martínez Muñoz.

Revisores: Luis Fernando Lago Fernández, Pedro Pablo Gómez Martín y Marco Antonio Gómez Martín.

Problema número 493

Altura final en Tetris

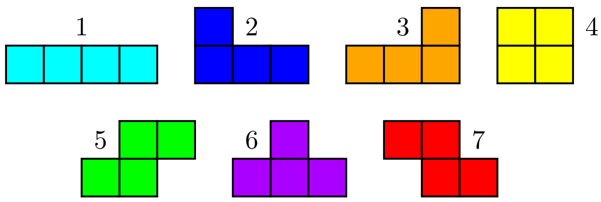
Tiempo máximo: 2,000 s   Memoria máxima: 4096 KiB

http://www.aceptaelreto.com/problem/statement.php?id=493

El Tetris es un videojuego que se popularizó en los años 80. Consiste en colocar una serie de piezas con distintas formas que van cayendo en un tablero, de tal modo que queden encajadas de la forma más compacta posible.

En este problema vamos a suponer una secuencia de piezas que caen, cada una en una determinada posición y con una determinada orientación que no se pueden cambiar. Las piezas se van amontonando según caen y no se eliminan las filas completas (como ocurre en el juego original). El objetivo es determinar la altura final de cada columna del tablero después de que caigan todas las piezas.

Hay un total de 7 piezas diferentes, mostradas en la figura:



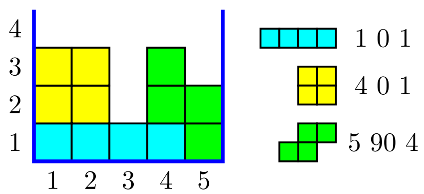
Entrada

La entrada está formada por distintos casos de prueba, cada uno ocupando varias líneas.

La primera línea contiene dos números, el número de columnas del tablero (C) y el número de piezas que van a caer (N). A continuación hay N líneas, cada una de ellas con la descripción de una pieza.

La descripción de una pieza consta de tres números: I, R y P. El primer número, I, es el identificador de la pieza (un número entre 1 y 7, en el mismo orden que en la figura). El segundo número, R, es la rotación de la pieza. Puede tomar los valores 0, 90, 180 o 270 y representa el ángulo de rotación de la pieza en el sentido contrario a las agujas del reloj. El tercer número, P, indica la posición de la pieza. Representa la columna más a la izquierda ocupada por la pieza. La numeración de las columnas empieza en 1.

Los valores mínimo y máximo para C y N son $4 \leq C \leq 100$, $1 \leq N \leq 100.000$. A modo de ejemplo, se muestra el resultado de colocar tres fichas en un tablero de 5 columnas.



El final de la entrada se indica con una línea con dos ceros que no se debe procesar.

Salida

Para cada caso de prueba, se escribirá una línea con la altura alcanzada en cada columna del tablero después de que caigan todas las piezas.

Entrada de ejemplo

```
5 3
1 0 1
4 0 1
5 90 4
8 4
6 270 4
1 180 5
1 90 6
7 0 4
0 0
```

Salida de ejemplo

```
3 3 1 3 2
0 0 0 9 9 8 3 3
```

Autor: Luis Fernando Lago Fernández.

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