

Olimpiada Boliviana de Informática
Fases: Distrital, Departamental y Final
2017



Olimpiada Boliviana de Informática

Recopilación de problemas por:
Samuel Réne Loza Ramírez
Los problemas corresponden a sus respectivos autores.

Índice

Alan y Matilde	1
Manu y su fiesta	2
Números Felices	3
Pares de unos	4
Tiempo de Ejecución	5
Notación O	7
Cuadrado	1
Factores Primos	2
Problema suma dos	3
Dar Cambio	4
Matriz Flip	5
Rotar Matriz	7
Fibonacci y Pitágoras	1
Sub Suma	2
Equilibrio	3
Cinta Multicolor	4
Juan y sus ovejas	6
Colores	7
Cinta Multicolor	9
Alan and his Knight	11

OBI - FASE DISTRITAL

Alan y Matilde

Matilde gusta de las matemáticas, y de hecho compite en las olimpiadas de matemáticas de su colegio; ella ha logrado batir a todos sus compañeros en un simple juego que tiene que ver con los factoriales.

Alan es un amigo de Matilde y apostó con ella que puede vencerle en su juego de factoriales. El día de mañana ellos se enfrentarán y Alan esta en problemas, necesita con urgencia que le ayuden a realizar un programa que sea capaz de vencer a Matilde!

El juego de los factoriales consiste en que dado un numero n , los jugadores deben calcular lo más rápidamente posible el dígito menos significativo (el de las unidades) del número $n!$. Matilde siempre ha sido la más veloz en responder y responde con exactitud.

Ahora los factoriales son fáciles de calcular, para un numero n , su factorial $n!$ se calcula multiplicando $1 * 2 * 3 * \dots * (n - 2) * (n - 1) * n$

En matemáticas el 0 en factorial es $0! = 1$, y se puede decir que:

$$0! = 1$$

$$1! = 1$$

$$2! = 1 * 2$$

$$3! = 1 * 2 * 3 \text{ y así sucesivamente.}$$

Puedes ayudar a que Alan consiga un programa lo suficientemente rápido en calcular el dígito que corresponde a las unidades para $n!$ siendo n un numero entero positivo?

Entrada

La primera línea contiene un número M entero que indica la cantidad de casos de prueba. A partir de la segunda línea se tienen los casos de prueba (uno por línea), para cada caso de prueba se tiene un único numero entero positivo ($n \leq 109$) del que se debe calcular la respuesta (el dígito de las unidades de $n!$).

Salida

Para cada caso de prueba imprimir la respuesta que consiste en el dígito de las unidades de $n!$, siendo n la entrada del caso de prueba.

Ejemplo de entrada

```
3
3
2
0
```

Ejemplo de salida

```
6
2
1
```

Manu y su fiesta

Manu tuvo una fiesta hace pocos días, y a cada uno de sus invitados por cortesía de Manu, se le entregaba una botella de vino. Sin embargo, antes de comenzar la fiesta uno de sus mayores enemigos había envenenado una de las botellas de su bodega, y antes de que la fiesta iniciara Manu hizo un experimento para saber cual era la botella envenenada.

Para encontrar la botella de vino envenenada, Manu pensó en utilizar r ratones que habían en su bodega; a los cuales les dio de beber de las botellas de vino. Se sabe que el veneno de las botellas tardaría en hacer efecto exactamente una hora, por lo que la prueba mortal la realizó una hora antes de que iniciara la fiesta. Al finalizar el experimento, Manu determinó que la botella b era la envenenada.

Manu ha dado de beber el vino de las botellas a una combinación única de ratones, razón por la cual ha sido posible encontrar la botella envenenada. A veces por la cantidad de ratones con los que se cuenta es imposible identificar la botella envenenada, utilizando los ratones. Sin embargo; Manu sabe... ! Esta parte de la historia ya no queremos saber, y menos cómo supo Manu cual es la botella envenenada!

Como te has dado cuenta, Manu utilizó ratones para saber cual es la botella envenenada, tu tarea es encontrar qué ratones habrían muerto en el intento de encontrar la botella de vino envenenada o si era imposible determinar la botella envenenada con los ratones.

Entrada

La primera línea contiene un número entero $1 \leq t \leq 10000$ que indica cuantas fiestas tuvo Manu. Las siguientes t líneas contienen dos números enteros b y r ; b es la botella envenenada y r es el número de ratones que tenía Manu antes de la fiesta.

Salida

Para cada fiesta, se pide imprimir el número de fiesta en una línea y en la siguiente línea, en caso de exista resultado usando los ratones, imprimir los ratones que Manu encuentra envenenados al inicio de la fiesta, separados por un espacio, o imprimir Imposible de otra manera.

Suponer que los ratones están enumerados de derecha a izquierda, si estos fueran colocados sobre una línea.

Ejemplo de entrada

```
5
7 4
10 2
1 2
8 3
8 4
```

Ejemplo de salida

```
Fiesta 1:3 2 1
Fiesta 2:Imposible
Fiesta 3:1
Fiesta 4:Imposible
Fiesta 5:4
```

Números Felices

Tome un número, eleve cada uno de sus dígitos al cuadrado. Finalmente sume los resultados parciales. Luego repite este proceso con la suma hasta obtener un dígito **1**. Veamos el ejemplo siguiente:

$$\begin{aligned}7^2 &= 49 \\4^2 + 9^2 &= 97 \\9^2 + 7^2 &= 130 \\1^2 + 3^2 + 0^2 &= 10 \\1^2 + 0^2 &= 1\end{aligned}$$

Cuando después de este proceso se llega a 1 se denomina número Feliz. Si el resultado no llega a 1 se denomina No Feliz.

Entrada

La entrada consiste de múltiples casos de prueba. La primera línea contiene el número de casos de prueba. Cada caso de prueba es un número $1 \leq n \leq 10^{14}$ en una línea.

Salida

En la salida escriba en una línea Feliz si el número es feliz y Triste si no es feliz.

Ejemplo de entrada

```
4
7
4
16
338
```

Ejemplo de salida

```
Feliz
Triste
Triste
Feliz
```

Pares de unos

Los números en su representación binaria están formados por unos y ceros. Se quiere conocer cuantos pares de unos seguidos existen en un número. Por ejemplo el número 7 en binario es 111 y existe un par. El número 3 tiene una pareja. El número 15 es el 1111 y tiene dos pares. El número 10 decimal no tiene pares de unos.

Dado un número decimal menor a 225 contar cuantos pares de unos existen en su representación binaria.

Entrada

La entrada consiste de varios casos de prueba, la primera línea contiene un número que indica el número de casos de prueba. Cada caso de prueba consiste en un número decimal menor a 2^{25}

Salida

Escriba en la salida el número de pares de unos que tiene el número.

Ejemplo de entrada

```
5
3
10
15
20
255
```

Ejemplo de salida

```
1
0
2
0
4
```


Tiempo de Ejecución

Near y Mello en la ciudad de La Paz, han comenzado a analizar bastante los algoritmos y sus diferentes tiempos de ejecución, en esta ocasión se requiere conocer automáticamente el tiempo de ejecución de un algoritmo dadas las líneas de código de un programa para ello tomar en cuenta las siguientes reglas de automatización.

- El tiempo de ejecución de un algoritmo es la suma total de los tiempos de ejecución de cada parte del algoritmo.
- Cualquier instrucción salvo un for tiene tiempo de ejecución 1
- Todo ciclo for tiene un tiempo de ejecución equivalente a n multiplicado por lo contenido dentro del ciclo

Entrada

La primera línea contiene un número entero $1 \leq t \leq 500$ que indica la cantidad de programas a analizar. Por cada programa, la primera línea tendrá un número entero $1 < c < 100$ que indica la cantidad de líneas de código del programa, en las siguientes c líneas se incluirán las líneas de código sin indentación en lenguaje C++.

Una línea de código puede ser:

```
Inicio de ciclo           : for(int cont = 0; cont < n; cont ++){
Fin de ciclo             : }
Declaración e Inicialización : int nombreVariable = 0;
Incremento de valor       : nombreVariable ++;
```

Debes considerar que el código es correcto.

Salida

Para cada programa imprimir el tiempo de ejecución de las líneas de código, según el siguiente formato **T(tiempo)**, donde **tiempo** es igual al tiempo de ejecución del algoritmo.

Ejemplo de entrada

```
4
6
for(int i=0;i<n;i++){
for(int j=0;j<n;j++){
for(int k=0;k<n;k++){
}
}
}
8
int x=0;
for(int i=0;i<n;i++){
}
for(int j=0;j<n;j++){
x++;
}
for(int k=0;k<n;k++){
}
6
for(int i=0;i<n;i++){
for(int j=0;j<n;j++){
}
}
for(int k=0;k<n;k++){
}
1
int x=0;
```

Ejemplo de salida

```
T(n(n(n)))
T(1+n+n(1)+n)
T(n(n)+n)
T(1)
```

Notación O

Recientemente ha salido un libro para los programadores competitivos, llamado Competitive Programmer's Handbook, en tal libro en el capítulo 2 se habla de complejidades y reglas para calcularlas según la Notación O, como está de moda automatizar todo, en esta ocasión se quiere automatizar el cálculo de complejidades y mostrar éstas en notación O.

Las reglas para el cálculo de complejidades en notación O son las siguientes:

- Cualquier instrucción salvo un for tiene complejidad 1
- Un for tiene complejidad n
- Si un for tiene contenido dentro uno o más for, la complejidad es equivalente a n elevado a la cantidad máxima de fors contenidos uno dentro de otro.

Entrada

La primera línea contiene un número entero $1 \leq t \leq 500$ que indica la cantidad de programas a analizar. Por cada programa, la primera línea tendrá un número entero $1 < c < 100$ que indica la cantidad de líneas de código del programa, en las siguientes c líneas se incluirán las líneas de código sin indentación en lenguaje C++.

Una línea de código puede ser:

```
Inicio de ciclo           : for(int cont = 0; cont < n; cont ++){
Fin de ciclo             : }
Declaración e Inicialización : int nombreVariable = 0;
Incremento de valor      : nombreVariable ++;
```

Salida

Para cada programa imprimir la complejidad en notación O de las líneas de código, según el siguiente formato **O(complejidad)**, donde **complejidad** es igual a la complejidad en notación O del programa, vea los ejemplos para mejor entendimiento

Ejemplo de entrada

```
4
6
for(int i=0;i<n;i++){
for(int j=0;j<n;j++){
for(int k=0;k<n;k++){
}
}
}
8
int x=0;
for(int i=0;i<n;i++){
}
for(int j=0;j<n;j++){
x++;
}
for(int k=0;k<n;k++){
}
6
for(int i=0;i<n;i++){
for(int j=0;j<n;j++){
}
}
for(int k=0;k<n;k++){
}
1
int x=0;
```

Ejemplo de salida

```
0(n^3)
0(n^1)
0(n^2)
0(1)
```

OBI - FASE DEPARTAMENTAL

Cuadrado

A usted le dan 4 palos cuyas longitudes son a, b, c, d . Así mismo existe una tienda para alterar el tamaño de los palos.

Por un peso pueden cambiar (incrementar o recortar) la longitud de cualquier palo en 1. Usted puede ir a la tienda las veces que desee. Su tarea es construir un cuadrado con los cuatro palos. Esto significa que los cuatro palos deben ser del mismo tamaño. Todas las longitudes deben ser positivas.

Por ejemplo si tenemos cuatro palos de longitudes 5, 6, 5, 5 la solución es recortar el palo de longitud 6 a 5, lo que da una respuesta de 1.

Entrada

La entrada consiste de múltiples casos de prueba. Cada caso de prueba consiste de los cuatro números a, b, c, d donde $1 \leq a, b, c, d \leq 1000000$.

La entrada termina cuando no hay más casos.

Salida

Por cada caso de prueba escriba en una línea el número mínimo de pesos requeridos para armar el cuadrado.

Ejemplo de entrada

```
5 6 5 5
1 2 3 4
1000 1000 1000 1000
```

Ejemplo de salida

```
1
4
0
```

Factores Primos

En teoría de números, los factores primos de un número entero son los números primos divisores exactos de ese número entero. El proceso de búsqueda de esos divisores se denomina factorización de enteros, o factorización en números primos.

Consideremos el número **24**, este puede descomponerse en $24 = 2 * 2 * 2 * 3 = 2^3 * 3$. Diremos que este número tiene dos factores que no se repiten.

Dados dos números A, B halle el número que tiene el mayor número de factores que no se repiten. Por ejemplo entre 10 y 50, tenemos los números 30 y 42 que son los que tienen el mayor número de factores y son 3. Lo que buscamos es el número más grande de estos dos que es el 42.

Entrada

La entrada consiste de múltiples casos de prueba. Cada caso de prueba consta de dos números $2 \leq A, B \leq 10^5$. La cantidad de casos de prueba no excede a 100 casos.

La entrada termina cuando no hay más datos.

Salida

Por cada caso de prueba escriba una línea con el número que tenga más factores primos, tal como se indicó y como se muestra en el ejemplo de salida.

Ejemplo de entrada

```
10 50
100 1000
10 200
```

Ejemplo de salida

```
42 tiene 3 factores
990 tiene 4 factores
198 tiene 3 factores
```

Problema suma dos

Dado un vector con números enteros se desea que encuentre dos números que sumen un valor S , Si hay más de dos pares de números imprima el que tenga el número más pequeño. Considere por ejemplo el vector:

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

Si la suma que buscamos es 9, la respuesta que debemos mostrar es **3** y **6**. Las dos soluciones son **(3,6)**,**(4,5)**. Dado que la primera solución comienza con el número más pequeño, ésta es la que buscamos.

Entrada

La entrada consiste de varios casos de datos de prueba. Cada caso de prueba consiste de dos líneas. La primera línea contiene dos números, el primero es el tamaño del vector ($n \leq 10^5$), y el segundo es la suma S .

La segunda línea contiene los n números del vector separados por un espacio.

La entrada termina cuando no hay más datos.

Salida

Imprima los dos números que sumen el valor buscado. Si no existe imprima -1

Ejemplo de entrada

```
9 9
1 3 2 6 4 1 2 3 5
```

Ejemplo de salida

```
3 6
```


Dar Cambio

Se están implementando máquinas para dar cambio. Las monedas disponibles en el país son de 1, 5, 10, 25, 50, y 100 centavos.

Las máquinas expendedoras de cambio tienen de 1 a 6 bolsillos para colocar monedas. Para reducir la cantidad de monedas que se utilizan al dar cambio, te piden calcular el mínimo de monedas requeridas.

Por ejemplo, si tenemos las 6 denominaciones, y hay que dar cambio de 289 pesos daremos 2 monedas de 100 centavos, 1 de 50 centavos, 1 de 25 centavos, 1 de 10 y 4 de 1 centavos, esto es $2 * 100 + 1 * 50 + 1 * 25 + 1 * 10 + 4 = 289$ que son un total de 9 monedas.

Las únicas denominaciones posibles son las anteriormente citadas.

Entrada

La entrada consiste en múltiples casos de prueba. Cada caso de prueba consta de una línea. El primer número ($N \leq 6$) es el número de denominaciones de monedas disponibles. Siguen N números separados por espacio y finalmente el monto de cambio $M \leq 1000$ que hay que dar. La entrada termina cuando no hay más datos.

Salida

Por cada caso de prueba escriba en la salida, una línea con el mínimo número de monedas que se requieren para dar cambio. Si no es posible dar cambio imprima "-1".

Ejemplo de entrada

```
6 1 5 10 25 50 100 289
1 100 300
3 1 5 25 100
3 100 50 5 89
```

Ejemplo de salida

```
9
3
4
-1
```

Matriz Flip

La entrada consiste de matrices cuadradas y el propósito es conocer que transformación sobre la matriz uno produjo la segunda matriz. Las transformaciones posibles son, flip horizontal, flip vertical o las combinaciones.

Dada la matriz:

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

Flip vertical es:

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

Flip horizontal es:

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

Entrada

Los datos consisten de múltiples casos de prueba. La primera línea de un caso de prueba consiste en el tamaño $N \leq 30$ de las dos matrices. Luego siguen N filas con N números enteros separados por un espacio, para la primera matriz. A continuación siguen N filas con N números enteros separados por un espacio, para la segunda matriz

La entrada termina cuando no hay más datos.

Salida

En la salida escriba según corresponda las transformaciones realizadas a la primera matriz para obtener la segunda.

```
flip vertical
flip horizontal
flip vertical y flip horizontal
sin transformaciones
respuesta desconocida
```

El orden de las respuestas es:

- **"sin transformaciones"** primera opcion.
- **"flip horizontal"** segunda opcion.
- **"flip vertical"** tercera opcion.
- **"flip vertical y flip horizontal"** cuarta opcion.
- **respuesta desconocida"** última opcion.

Ejemplo de entrada

```
6 1 5 10 25 50 100 289
1 100 300
3 1 5 25 100
3 100 50 5 89
```

Ejemplo de salida

```
9
3
4
-1
```

Rotar Matriz

La entrada consiste de matrices cuadradas y el propósito es el de rotar la matriz 90, 180, o 270 grados. La rotación de una matriz en 360 grados volvería a la matriz original.

Por ejemplo, si tenemos la matriz:

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

el resultado después de rotar 90 grados es:

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

Entrada

La entrada consiste de varios casos de prueba. La primera línea de un caso de prueba consiste en el tamaño $N \leq 30$ de la matriz y los grados que hay que **rotar(90,180,270)**. Luego siguen N filas con N números enteros separados por un espacio.

La entrada termina cuando no hay mas datos.

Salida

En la salida escriba la matriz rotada la cantidad de grados solicitado. Al imprimir la matriz tenga el cuidado de no dejar un espacio al final de la línea.

Ejemplo de entrada

```
4 90
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

Ejemplo de salida

```
13 9 5 1
14 10 6 2
15 11 7 3
16 12 8 4
```

OBI - FASE NACIONAL

Fibonacci y Pitágoras

El famoso teorema de Pitágoras indica que la hipotenusa c de un triángulo rectángulo de lados a, b se puede calcular con la fórmula $c^2 = a^2 + b^2$. Este teorema se ha hecho tan famoso que muchos números se han denominados pitagóricos si pueden hallarse como la suma de dos números enteros elevados al cuadrado.

¿Qué tiene que ver Fibonacci con Pitágoras?. Bien primero recordemos que la sucesión de Fibonacci se define Matemáticamente con la ecuación $f(n) = f(n - 1) + f(n - 2)$. Los primeros números de la serie son: 0, 1, 1, 2, 3, 5, 8, 13..etc.

Tratando de ver si se pueden encontrar números pitagóricos formados por exclusivamente números de Fibonacci, se ve lo siguiente: El primer número pitagórico que se puede hallar es el numero $1 = 0^2 + 1^2$, el segundo es el $2 = 1^2 + 1^2$, así se forma una secuencia de números que son de Fibonacci y pitagóricos, simultáneamente. Los primeros elementos de esta serie son: 1, 2, 5, 13.... todos formados exclusivamente con números de Fibonacci.

Entrada

La entrada consiste de varios casos de prueba. La primera linea indica cuantos casos de prueba existen. Cada caso de prueba contiene un numero, que indica la posición en la secuencia de los números Fibonacci pitagóricos que queremos hallar.

Salida

Imprima en la salida el número Pitagorico indicado.

Ejemplo de entrada

```
4
0
1
2
3
```

Ejemplo de salida

```
1
2
5
13
```

Sub Suma

Dado un vector \mathbf{V} se quiere hallar la suma de un grupo de elementos contiguos cuya suma sea un valor dado \mathbf{S} . No se puede re ordenar el vector.

Por ejemplo si se tiene un vector con 6 elementos:

1	2	3	4	5	6
2	8	2	6	3	5

Ahora se quiere hallar cuales son los elementos cuya suma es 16, se observa que los elementos 2, 3, 4 suman $16(8 + 2 + 6 = 16)$. También los elementos $2 + 6 + 3 + 5 = 16$. Sin embargo la respuesta correcta es 2, 4 que significa que debemos sumar los valores desde la posición 2 hasta la 4 inclusive.

Entrada

La entrada consiste de varios datos de prueba. Cada caso de prueba consiste de dos líneas. La primera línea tiene dos números, el primero es el número de elementos del vector ($0 \leq n \leq 107$) y el segundo es la suma ($1 \leq s \leq 1014$) que de elementos contiguos que buscamos.

La segunda línea contiene los n elementos del vector separados por un espacio.

Los datos de entrada terminan cuando no hay mas datos.

Salida

En la salida imprima dos números que representan la posición del primer elemento y el último elemento del arreglo, cuya suma de los elementos entre estos valores es la suma buscada. En caso de que no exista imprima -1.

Ejemplo de entrada

```
6 16
2 8 2 6 3 5
7 5
1 4 4 1 0 0 5
8 6
7 1 0 9 4 9 2 6
9 32
7 9 3 6 4 3 2 5 2
10 4
2 5 7 5 8 4 9 2 6 9
```

Ejemplo de salida

```
2 4
1 2
8 8
1 6
6 6
```

Equilibrio

Los números tienen varias propiedades, una de ellas es la primalidad, es así que todos los primos son equilibrados en sus factores, ya que el factor primo menor y mayor que los divide es el mismo número. Pero lo curioso es que hay algunos números que también son equilibrados sin ser primos; por ejemplo el 81, su factor primo menor y mayor es 3, ya que no tiene otro factor primo.

Tu tarea es, decidir si un número tiene equilibrio, caso contrario indicar el factor primo menor y mayor del número.

Entrada

La primera línea contiene un número entero positivo c que indica cuántos casos de prueba existirán.

En seguida se tienen c líneas, cada una tiene n que es el número del cual se quiere saber si tiene equilibrio o no.

Salida

Para cada caso se debe imprimir en una línea el factor mínimo y el factor máximo del número en caso de no tener equilibrio; caso contrario se debe imprimir Equilibrio.

El formato de la salida para los números que no tienen equilibrio es $[factorMin, factorMax]$.

Ejemplo de entrada

```
9
5
27
87
80
45
60
90
42
52
```

Ejemplo de salida

```
Equilibrio
Equilibrio
[3,29]
[2,5]
[3,5]
[2,5]
[2,5]
[2,7]
[2,13]
```


Cinta Multicolor

Está de moda las cosas multi, y en esta ocasión se tienen las cintas multicolor, estas cintas tienen una peculiaridad y es que nueve colores están presentes en la misma. Vienen en diferentes anchos y los colores se presentan de forma sesgada y cada color está representado por un dígito.

Un ejemplo se observa a continuación:

1111
1112
1122
1222
2222
2223
2233
2333
3333
3334
3344
3444
4444
4445
4455
4555
5555
5556
5566
5666
6666
....
....

Esta es una cinta de ancho 4, para efectos de nuestro problema la cinta puede ser tan larga como te imagines.

Marita se está preguntando que colores estarán presentes si corta la cinta de largo n . En el ejemplo si se hace un corte de largo 12 los colores en los que termina la cinta serán 3444 pero si se corta de una longitud 41 los colores son 2222. Pues la tarea es ayudar a Marita a saber en que colores termina la cinta si se tiene un corte de largo n .

Entrada

Se tiene varios casos de prueba, cada caso de prueba se encuentra en una línea que consta de dos valores a y c el ancho y el largo de corte de la cinta respectivamente.

Los casos terminan cuando ya no hay más datos.

Salida

Por cada caso de prueba debes imprimir en una línea los colores en los que termina la cinta.

Juan y sus ovejas

Juan tiene muchas ovejas. Cada oveja tiene un arete con un código que la identifica, estos códigos comienzan en 1 y llegan al número de ovejas que tiene. No hay códigos repetidos. Quiere determinar cuantas razas diferentes tiene. Pero como no conoce mucho de razas, lo que decidió hacer es tomar pares de ovejas que él cree pueden ser de la misma raza y registrar sus códigos.

Por ejemplo si, tiene 9 ovejas y registro 6 pares

(4, 7), (5, 3), (2, 5), (8, 6), (1, 5), (4, 5)

vea que los pares (4, 7), (5, 3), (2, 5), (1, 5), (4, 5) corresponden a la misma raza. El par (8, 6) es una raza diferente y la oveja 9 no fue considerada por lo que también corresponderá a otra raza.

Se aprecia que hay tres razas y la raza que más ovejas tiene es 6.

Entrada

La entrada consiste de varios casos de prueba. Cada caso de prueba consiste de dos números en una línea separados por un espacio. El primero es el número de ovejas y el segundo el numero de pares de ovejas que registro. La entrada termina con 0 0.

Salida

Imprima en la salida el número de razas y el numero de ovejas de la raza que más ovejas tiene.

Ejemplo de entrada

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

Ejemplo de salida

```
Existen 5 posibles razas
La raza que tiene mas ovejas tiene 2
Existen 3 posibles razas
La raza que tiene mas ovejas tiene 6
```

Colores

Melchor ha sido contratado por el virrey como pintor oficial del virreinato, frecuentemente el Virrey solicita a Melchor pintar muchos cuadros, como Melchor es un habilidoso y eficiente pintor esta tarea no le resulta complicada, sin embargo al virrey le gustan solo colores de Mersenne en los cuadros que pide, dada una dimensión de n centímetros de alto y m centímetros de ancho de un cuadro a pintar.

Para ello Melchor usa una paleta de varios colores los mismos que están identificados como números, donde el 1 es el color blanco que nunca es pintado.

El cuadro puede ser visto como una cuadrícula de $n \times m$ cuadraditos cada uno de ellos de $1cm$. Por lo que, si se tiene un cuadro de 3×4 , la cuadrícula se enumera de izquierda a derecha y de arriba a abajo, empezando en 0; como se muestra en el Cuadro 1.

0	1	2	3
4	5	6	7
8	9	10	11

Cuadro 1: Ejemplo de ubicación de cuadrículas.

Como a Melchor le gustan los cuadros coloridos, a él le gustaría pintar cada espacio de la cuadrícula con los colores desde 1 hasta C_n en orden creciente; sin embargo eso no es posible para no desperdiciar pintura. Para ello Melchor utiliza el hecho de que los colores se comportan de una manera particular.

Si se quiere pintar la celda n se utiliza los colores de 1 a C_n , y para determinar el color de la celda se utiliza el dominio de colores sobre colores, de tal manera que los colores para cada posición i dominan al color $i + 1$ y así hasta C_n circularmente, esto quiere decir que el color del final puede dominar al color del inicio si se los ordenase en una línea.

Por ejemplo, si se tuviera que pintar la quinta cuadrícula del cuadro, se usan los siguientes colores $\{1,2,3,4,5\}$, como Melchor pinta con todos los colores posibles, estos colores cambiarían por la ley de dominio de colores de su estado inicial a $\{1,3,4,5\}$ cuando 1 domina a 2, $\{1,3,5\}$ cuando 3 domina a 4, $\{3,5\}$ cuando 5 domina a 1, y finalmente $\{3\}$ cuando 3 domina a 5. Por lo tanto, para la posición 5 del cuadro, Melchor sólo debe usar el color 3 para no desperdiciar pintura. Y además sólo usa colores de Mersenne, un color de Mersenne se identifica con un número que se obtiene con la fórmula $2^k - 1$, para algún k positivo.

Tu tarea es ayudar a Melchor a determinar que colores usar en las obras que el virrey le encarga, y mostrar el cuadro resultante. Una característica de Melchor es que en la esquina superior izquierda del cuadro pone su firma artística la cual es mostrada como una ' M '.

Recordar que cuando Melchor no puede pintar un espacio del cuadro este es mostrado como un asterisco '*', esto sucede cuando el color dominante es el blanco o cuando no es un color de Mersenne.

Entrada

La entrada consiste de varios casos de prueba. Cada caso de prueba consiste de dos números en una línea separados por un espacio. El primero es el número de ovejas y el segundo el número de pares de ovejas que registro. La entrada termina con 00.

Salida

Imprima en la salida el número de razas y el número de ovejas de la raza que más ovejas tiene.

Ejemplo de entrada

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

Ejemplo de salida

```
Existen 5 posibles razas
La raza que tiene mas abejas tiene 2
Existen 3 posibles razas
La raza que tiene mas abejas tiene 6
```

Cinta Multicolor

Está de moda las cosas multi, y en esta ocasión se tienen las cintas multicolor, estas cintas tienen una peculiaridad y es que nueve colores están presentes en la misma. Vienen en diferentes anchos y los colores se presentan de forma sesgada y cada color está representado por un dígito. Un ejemplo se observa a continuación:

1111
1112
1122
1222
2222
2223
2233
2333
3333
3334
3344
3444
4444
4445
4455
4555
5555
5556
5666
6666
....
....

Esta es una cinta de ancho 4, para efectos de nuestro problema la cinta puede ser tan larga como te imagines.

Marita se está preguntando que colores estarán presentes si corta la cinta de largo n . En el ejemplo si se hace un corte de largo 12 los colores en los que termina la cinta serán 3444 pero si se corta de una longitud 41 los colores son 2222. Pues la tarea es ayudar a Marita a saber en que colores termina la cinta si se tiene un corte de largo n

Entrada

Se tiene varios casos de prueba, cada caso de prueba se encuentra en una línea que consta de dos valores a y c el ancho y el largo de corte de la cinta respectivamente.

Los casos terminan cuando ya no hay más datos.

Salida

Por cada caso de prueba debes imprimir en una línea los colores en los que termina la cinta.

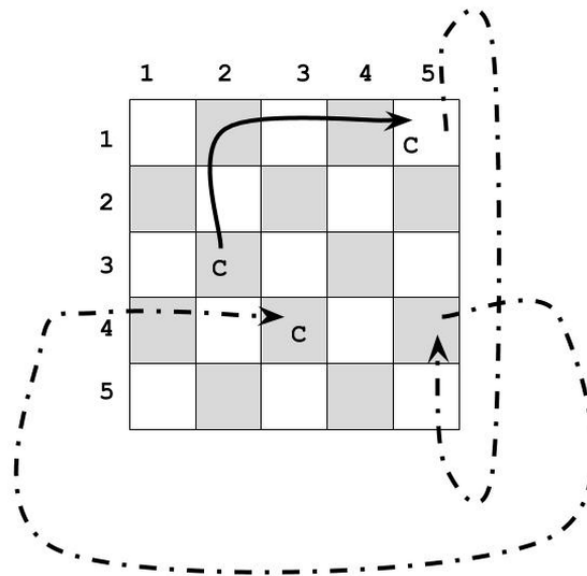
Alan and his Knight

Alan ha tomado cursos de Ajedrez las últimas vacaciones invernales, y en los recreos ha estado jugando con sus amigos algunas partidas rápidas; Su compañero Diego está furioso porque nunca ha podido ganarle. Diego ha escuchado del profe de Computación un peculiar Ajedrez extendido en el que las fichas atraviesan los límites del tablero y continúan su movimiento en el borde opuesto. En sus típicos arranques de furia, Diego ha retado a Alan a resolver un problema en el Tablero extendido: Si ponemos en el tablero un Caballo en la posición (f,c) , y a partir de esta posición el Caballo comienza a moverse en una de las ocho posibles variantes: Arriba Izquierda (AI), Arriba Derecha (AD), Abajo Izquierda (BI), Abajo Derecha (BD), Derecha Arriba (DA), Derecha Abajo (DB), Izquierda Arriba (IA), Izquierda abajo (IB). En su trayecto el Caballo es posible que alcance una casilla en la que estuvo antes. ¿Cuál es el mínimo número de movidas que el caballo puede realizar hasta alcanzar una casilla en la que ya estuvo?

Diego se ha memorizado las posibilidades para cada posición del caballo y está seguro de ganar contra Alan. Conociendo que Diego propone el reto, Alan pide que el tablero extendido pueda variar en tamaño. El tablero puede ser de $NF \times NC$ donde NF es el número de filas y NC es el número de columnas. El Caballo podrá moverse en la dirección escogida df casillas en las filas y dc casillas en las columnas. Así ambos tendrán que calcular la respuesta sin ventajas. El tablero se enumera desde la casilla superior izquierda; las filas hacia abajo y las columnas hacia la derecha.

Por ejemplo, si se tiene un tablero de 5×5 , y el caballo se halla en la posición $(3,2)$ con desplazamientos en filas y columnas $df = 2$ y $dc = 3$; si el caballo se mueve AD acabará en la casilla $(1,5)$, como ya no le es posible cambiar de dirección, continuará moviendo AD, luego a partir de $(1,5)$ alcanzará la casilla $(4,3)$.

Diego está desesperado y sabe que Alan es muy bueno calculando. Ayuda a Diego a conseguir un programa que calcule rápidamente los resultados y así pueda ganar el reto. ¡Diego no sabe programar, depende de ti!



Entrada

La primera línea contiene un número M entero que indica la cantidad de casos de prueba. A partir de la segunda línea se tienen los casos de prueba (uno por línea), para cada caso de prueba se tiene seis números enteros positivos NF , NC , F , C , DF , DC y DIR . Donde NF y NC ($1 \leq NF, NC \leq 10000000$) representan el número de filas y columnas del tablero respectivamente; F y C ($1 \leq F \leq NF$ y $1 \leq C \leq NC$) es la posición inicial del caballo en fila y columna respectivamente; con DF y DC ($1 \leq DF < NF$ y $1 \leq DC < NC$) el desplazamiento en fila y columna que el caballo debe realizar, DIR es la dirección que el Caballo seguirá ($AI, AD, BI, BD, DA, DB, IA, IB$).

Salida

Para cada caso de prueba imprimir la respuesta que consiste en el mínimo número de movidas que el caballo puede realizar hasta alcanzar una casilla en la que ya estuvo.

Ejemplo de entrada

```
2
5 5 3 2 2 3 AI
6 5 2 2 1 5 DB
```

Ejemplo de salida

```
5
6
```