**Problema: matriz mínima de productos**

La tarea es encontrar la suma mínima de productos de dos matrices del mismo tamaño, dado que se permiten k modificaciones en la primera matriz. En cada modificación, un elemento de la matriz de la primera matriz puede aumentarse o disminuirse en 2.   
  
Nota : la suma del producto es la suma (A [i] \* B [i]) para todo i de 1 a n donde n es el tamaño de ambas matrices

**Formato de entrada:**

1. La primera línea de la entrada contiene n y k delimitados por espacios en blanco
2. La segunda línea contiene la matriz A (matriz modificable) con sus valores delimitados por espacios
3. La tercera línea contiene la matriz B (matriz no modificable) con sus valores delimitados por espacios

**Formato de salida:**   
  
genera la suma mínima de productos de las dos matrices

**Restricciones:**

1. **1 ≤ N ≤ 10 ^ 5**
2. **0 ≤ | A [i] |, | B [i] | ≤ 10 ^ 5**
3. **0 ≤ K ≤ 10 ^ 9**

**Muestra de entrada y salida**

|  |  |  |
| --- | --- | --- |
| **SNo.** | **Entrada** | **Salida** |
| 1 | 3 5  1 2 -3  -2 3 -5 | -31 |
| 2 | 5 3  2 3 4 5 4  3 4 2 3 2 | 25 |

**Explicación para la muestra 1:**   
  
Aquí los números totales son 3 y las modificaciones totales permitidas son 5. Entonces modificamos A [2], que es -3 y lo aumentamos 10 (ya que se permiten 5 modificaciones). Ahora la suma final será   
(1 \* -2) + (2 \* 3) + (7 \* -5)   
-2 + 6 - 35   
-31   
  
-31 es nuestra respuesta final.   
  
**Explicación para la muestra 2:**   
  
Aquí los números totales son 5 y las modificaciones totales permitidas son 3. Entonces modificamos A [1], que es 3 y lo disminuimos en 6 (ya que se permiten 3 modificaciones).   
Ahora la suma final será   
(2 \* 3) + (-3 \* 4) + (4 \* 2) + (5 \* 3) + (4 \* 2)   
6 - 12 + 8 + 15 + 8   
25   
  
25 es nuestra respuesta final. 

***Nota*** :  
  
*Por favor, no use el paquete y el espacio de nombres en su código. Para lenguajes orientados a objetos, su código debe escribirse en una clase.*

***Nota*** : Los  
  
*participantes que envían soluciones en lenguaje C no deben usar funciones de <conio.h> / <process.h>, ya que estos archivos no existen en gcc.*

***Nota*** :  
  
*Para C y C ++, el tipo de retorno de la función main () debe ser int.*

**© 2016 Tata Consultancy Services Limited. Todos los derechos reservados.**

**Problema: Pirámide Lógica**

Identifique la lógica detrás de la serie   
  
6 28 66 120 190 276 ....   
  
Los números de la serie se deben usar para crear una Pirámide. La base de la pirámide será la más ancha y comenzará a converger hacia la parte superior donde solo habrá un elemento. Cada capa sucesiva tendrá un número menor que el de la capa debajo de él. El ancho de la pirámide se especifica mediante un parámetro de entrada N. En otras palabras, habrá N números en la capa inferior de la pirámide.   
  
Las reglas de construcción de Pyramid son las siguientes

1. El primer número de la serie debe estar en la parte superior de la pirámide
2. Último número N de la serie debe estar en la capa más inferior de la pirámide, con N º número es la más a la derecha el número de esta capa.
3. Los números de menos de 5 dígitos deben rellenarse con ceros para mantener la santidad de una pirámide cuando se imprimen. Eche un vistazo a los ejemplos a continuación para obtener una comprensión gráfica de lo que realmente significa esta regla.

Ejemplo   
  
**Si la entrada es 2, la salida será**   
  
 00006   
00028 00066   
  
**Si la entrada es 3, la salida será**   
  
  00006   
 00028 00066   
00120 00190 00276   
  
Las especificaciones de entrada y salida formales se detallan a continuación

**Formato de entrada: la**   
  
primera línea de entrada contendrá el número N que corresponde al ancho de la capa más inferior de la pirámide.

**Formato de salida:**   
  
la pirámide construida a partir de números en la serie según las reglas de construcción establecidas

**Restricciones:**

1. **0 <N <= 14**

**Muestra de entrada y salida**

|  |  |  |
| --- | --- | --- |
| **SNo.** | **Entrada** | **Salida** |
| 1 | 2 | https://www.tcscodevita.com/CodevitaV6/images/pyramid2.png |
| 2 | 3 | https://www.tcscodevita.com/CodevitaV6/images/pyramid1.png |

***Nota*** :  
  
*Por favor, no use el paquete y el espacio de nombres en su código. Para lenguajes orientados a objetos, su código debe escribirse en una clase.*

***Nota*** : Los  
  
*participantes que envían soluciones en lenguaje C no deben usar funciones de <conio.h> / <process.h>, ya que estos archivos no existen en gcc.*

***Nota*** :  
  
*Para C y C ++, el tipo de retorno de la función main () debe ser int.*

**© 2016 Tata Consultancy Services Limited. Todos los derechos reservados.**

**Problema: suma principal consecutiva**

Algunos números primos se pueden expresar como la suma de otros números primos consecutivos.   
Por ejemplo   
  
5 = 2 + 3   
17 = 2 + 3 + 5 + 7   
41 = 2 + 3 + 5 + 7 + 11 + 13   
  
Su tarea es averiguar cuántos números primos que satisfacen esta propiedad están presentes en el rango de 3 a N sujeto a una restricción que la suma siempre debe comenzar con el número 2.   
  
Escriba el código para averiguar el número de números primos que satisfacen la propiedad mencionada anteriormente en un rango dado.

**Formato de entrada: la**   
  
primera línea contiene un número N

**Formato de salida:**   
  
Imprime el número total de todos esos números primos que son menores o iguales que N.

**Restricciones:**

**1. 2 <N <= 12,000,000,000**

**Muestra de entrada y salida**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNo.** | **Entrada** | **Salida** | **Comentario** |
| 1 | 20 | 2 | (Debajo de 20, hay 2 de esos números: 5 y 17).  5 = 2 + 3  17 = 2 + 3 + 5 + 7 |
| 2 | 15 | 1 |  |

***Nota*** :  
  
*Por favor, no use el paquete y el espacio de nombres en su código. Para lenguajes orientados a objetos, su código debe escribirse en una clase.*

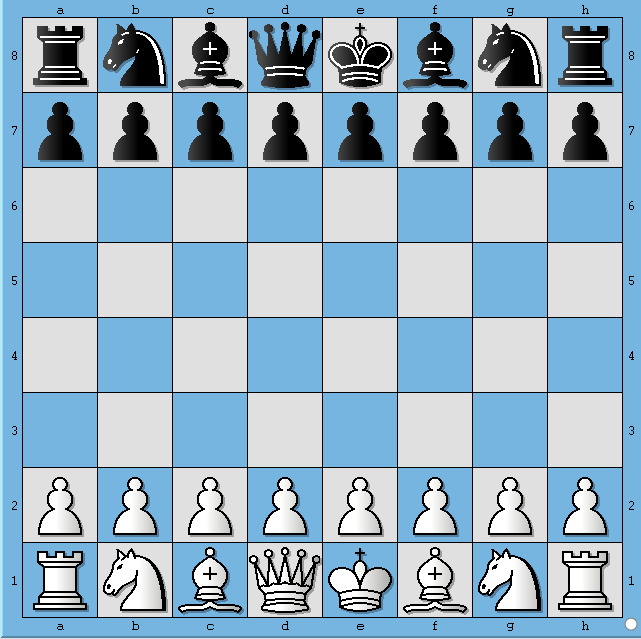
***Nota*** : Los  
  
*participantes que envían soluciones en lenguaje C no deben usar funciones de <conio.h> / <process.h>, ya que estos archivos no existen en gcc.*

***Nota*** :  
  
*Para C y C ++, el tipo de retorno de la función main () debe ser int.*

**© 2016 Tata Consultancy Services Limited. Todos los derechos reservados.**

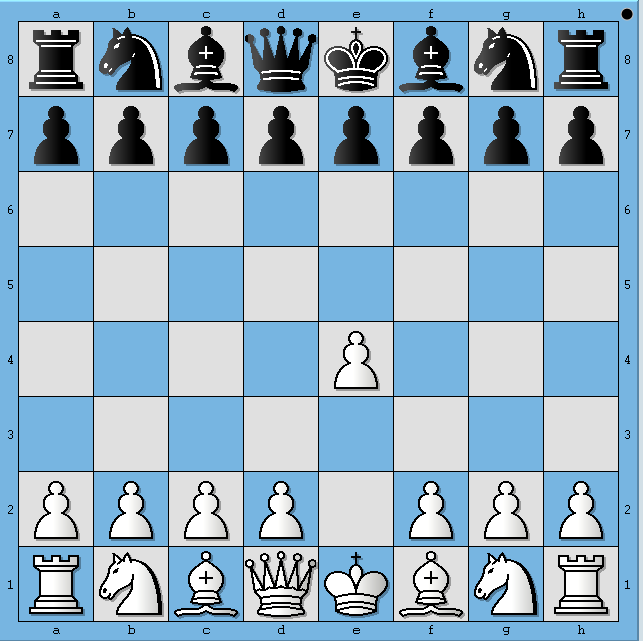
**Problema: Bishop Moves**

**Fondo**  
  
Una posición de tablero de ajedrez se captura con precisión mediante la notación Forsyth-Edwards y se abrevia como FEN. Un "registro" de FEN define una posición de juego particular, todo en una sola línea de texto y utilizando solo el juego de caracteres ASCII. Un registro FEN contiene seis campos. Puede encontrar una descripción completa del formato FEN para representar posiciones de Ajedrez [aquí.](https://en.wikipedia.org/wiki/Forsyth%E2%80%93Edwards_Notation)   
  
Para el propósito de este problema, solo considere el primero de los seis campos de FEN. Antes de describir el problema, veamos cómo FEN se mapea a una posición de tablero. Las siguientes 5 imágenes muestran las posiciones de la placa y su correspondiente representación FEN.

   
                    **Figura 1.**

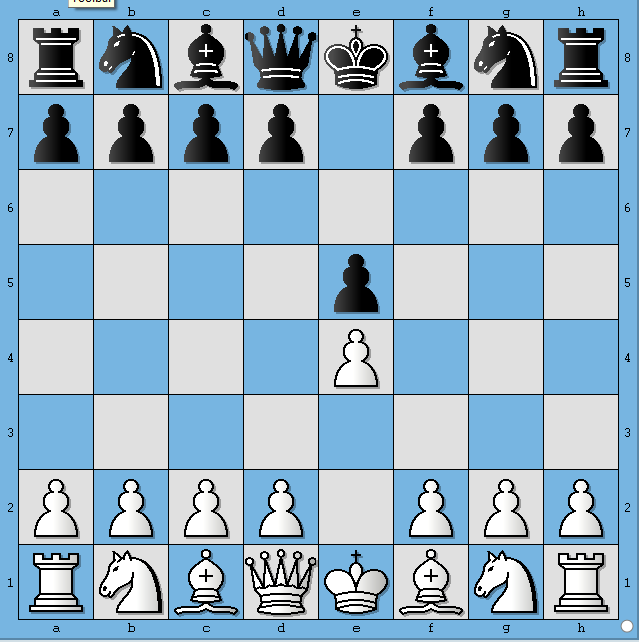
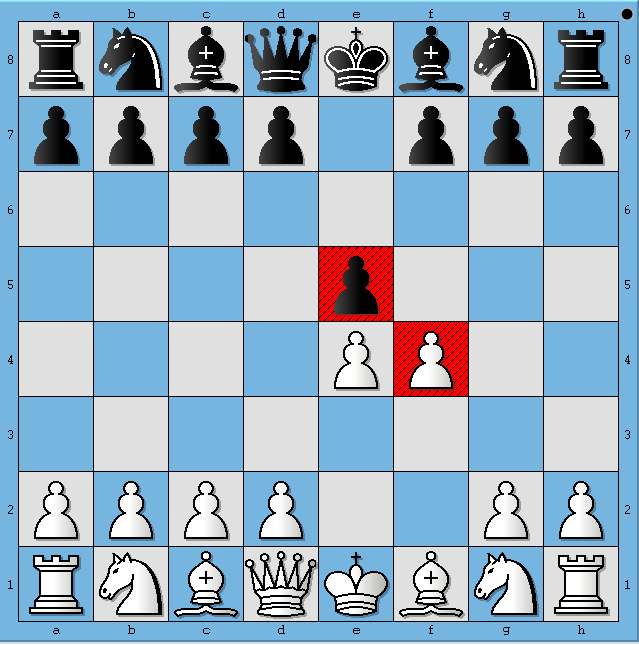
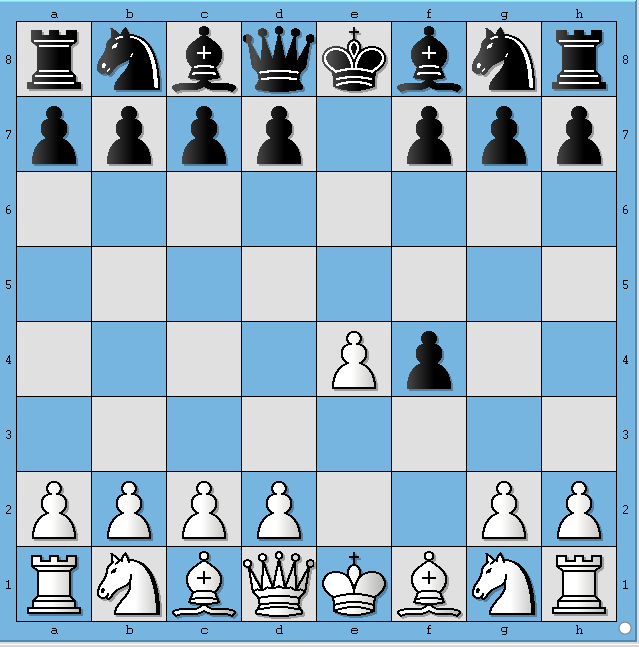
|  |
| --- |
| Esta posición de la tabla representa la posición inicial antes de que cualquier lado haya realizado un movimiento.En formato FEN, esta posición de la placa se representa como    rnbqkbnr / pppppppp / 8/8/8/8 / PPPPPPPP / RNBQKBNR w |

Digamos que White juega e4. Entonces la posición de la tabla se ve como se muestra a continuación 

   
                    **Figura 2.**

|  |
| --- |
| Esta posición de la tabla representa el tablero de ajedrez después de que las blancas hayan jugado e4. En formato FEN, esta posición de la placa se representa como    rnbqkbnr / pppppppp / 8/8 / 4P3 / 8 / PPPP1PPP / RNBQKBNR b |

Del mismo modo, se representan 3 medios movimientos más en los siguientes diagramas

                         **Figura 3. Figura 4. Figura 5.**

Los FEN correspondientes a las Figuras 3, 4 y 5 se representan como   
  
           3. rnbqkbnr / pppp1ppp / 8 / 4P3 / 4P3 / 8 / PPPP1PPP / RNBQKBNR w   
           4. rnbqkbnr / pppp1ppp / 8 / 4p3 / 4PP2 / 8 / PPPP2PP / RNBQKBNR b   
           5 . rnbqkbnr / pppp1ppp / 8/8 / 4Pp2 / 8 / PPPP2PP / RNBQKBNR w   
  
**Wikipedia describe el primer campo del formato FEN de la siguiente manera**   
  
Se describe cada rango, comenzando con el rango 8 y terminando con el rango 1; dentro de cada rango, los contenidos de cada cuadrado se describen desde el archivo "a" hasta el archivo "h". Siguiendo la [notación algebraica estándar](https://en.wikipedia.org/wiki/Algebraic_chess_notation)(SAN), cada pieza se identifica con una sola letra tomada de los nombres estándar en inglés (peón = "P", caballero = "N", alfil = "B", torre = "R", reina = "Q" y rey = "K"). [1] Las piezas blancas se designan con letras mayúsculas ("PNBRQK"), mientras que las negras se usan en minúsculas ("pnbrqk"). Los cuadrados vacíos se anotan usando los dígitos del 1 al 8 (el número de cuadrados vacíos), y "/" separa los rangos.   
  
El segundo campo denota de quién es el movimiento ahora. "w" representa que es el turno de las Blancas para jugar y "b" indica que es el turno de las negras de jugar la   
  
**Declaración del Problema de CodeVita**   
  
Dada la posición de la Junta en formato FEN, su tarea es descubrir todos los movimientos que Bishop (s) ) del lado de juego puede hacer.

**Formato de entrada:**

1. La primera línea contiene un solo registro FEN, que corresponde a una posición particular de la placa y también indica de quién es el turno.

**Formato de salida:**

1. La salida debe imprimirse de la siguiente manera
   1. Todos los movimientos legales que pueda realizar Bishop deben tener el formato "[<Mover formato>]"
   2. Donde <Mover formato> se mueve representado en formato "[fromSquare] [toSquare]"
2. Vea la sección Ejemplo para una mejor comprensión del formato de salida
3. Siga la especificación de impresión de salida para imprimir la salida en el formato requerido

**Restricciones:**

1. Dado que nos centramos solo en las dos primeras partes del FEN, esencialmente estamos ignorando la posibilidad de enroque. **Por lo tanto, nuestros casos de prueba no contienen FEN que den lugar a tales posiciones.**

**Muestra de entrada y salida**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNo.** | **Entrada** | **Representación del tablero** | **Salida** |
| 1 | 3k4 / 8/8 / 2P1P3 / 3B4 / 2R1R3 / 8 / 4K3 w | https://www.tcscodevita.com/CodevitaV6/images/ex1.png | [] |
| 2 | 3k4 / 8/8 / 4P3 / 3B4 / 2R1R3 / 8 / 4K3 w | https://www.tcscodevita.com/CodevitaV6/images/ex2.png | [d4a7, d4b6, d4c5] |
| 3 | 3k4 / 8/8/8 / 3B4 / 8/8 / 1B2K3 w | https://www.tcscodevita.com/CodevitaV6/images/ex3.png | [d4h8, d4a7, d4g7, d4b6, d4f6, d4c5, d4e5, d4c3, d4e3, d4b2, d4f2, d4a1, d4g1, b1h7, b1g6, b1f5, b1e4, b1d3, b1a2, b1c2] |

**Especificación de impresión:**

1. Debería comenzar con "[" y terminar con "]"
2. Si es posible más de un movimiento, los movimientos deben estar separados por una coma seguida de espacios en blanco
3. Los movimientos de un solo obispo se deben imprimir en *Formato de movimiento* . Escanee el pizarrón desde el 8 ° rango al 1 ° rango desde un archivo a un archivo h. Cualquiera que sea el cuadrado que se golpee primero, ese movimiento debe imprimirse primero.
4. Si existe más de un alfil para que se mueva el lateral, entonces comience a escanear el alfil desde el 8 ° rango al 1 ° rango, de izquierda a derecha, es decir, desde un archivo a un archivo h. Sea cual sea el obispo que aparezca primero, imprima todos los movimientos para ese obispo primero.
5. Verifique su comprensión de cómo debe ocurrir la impresión en comparación con los ejemplos que se muestran arriba

***Nota*** :  
  
*Por favor, no use el paquete y el espacio de nombres en su código. Para lenguajes orientados a objetos, su código debe escribirse en una clase.*

***Nota*** : Los  
  
*participantes que envían soluciones en lenguaje C no deben usar funciones de <conio.h> / <process.h>, ya que estos archivos no existen en gcc.*

***Nota*** :  
  
*Para C y C ++, el tipo de retorno de la función main () debe ser int.*

**© 2016 Tata Consultancy Services Limited. Todos los derechos reservados.**

**Problema: Tabla de la Liga de Fútbol**

**Declaración:**   
  
todas las principales ligas de fútbol tienen mesas de Grandes Ligas. Cada vez que se juega un nuevo partido, la tabla de la liga se actualiza para mostrar las clasificaciones actuales (en función de los puntajes, las metas para (GF), las metas en contra (GA)). Teniendo en cuenta los resultados de algunos partidos entre los equipos, escriba un programa para imprimir todos los nombres de los equipos en orden ascendente (Líder en la parte superior y Rezagado en la parte inferior) en función de sus clasificaciones.   
  
**Reglas: :**

1. Una victoria resulta en 2 puntos, un empate resulta en 1 punto y una pérdida vale 0 puntos.
2. El equipo con más goles en un partido gana el partido.
3. La diferencia de goles (GD) se calcula como Metas para (GF) - Metas en contra (GA)
4. La clasificación se decide de la siguiente manera
   1. El equipo con puntos máximos se clasifica en el puesto 1 y los puntos mínimos se colocan en último lugar
   2. Los vínculos se rompen de la siguiente manera
      1. Los equipos con los mismos puntos se clasifican según la diferencia de objetivos (GD).
      2. Si la Diferencia de goles (GD) es la misma, entonces el equipo con Goles mayores se clasifica por delante
      3. Si GF son iguales, los equipos deben estar en el mismo rango, pero deben imprimirse en alfabético insensible a mayúsculas de acuerdo con los nombres del equipo.
5. Los equipos pueden jugar un máximo de dos partidos uno contra el otro - partidos de local y visitante respectivamente
6. Más de 2 partidos de los mismos equipos, deben considerarse como Invalid Input
7. Un equipo no puede jugar partidos contra sí mismo, por lo tanto, si los nombres de los equipos son los mismos para un partido determinado, debe considerarse como Entrada no válida.

**Formato de entrada: la**   
  
primera línea de entrada contendrá el número de equipos (N) La   
segunda línea contiene los nombres de los equipos (Na) delimitados por un carácter en blanco La   
tercera línea contiene el número de coincidencias (M) cuyos resultados están disponibles Las   
siguientes líneas M contienen una coincidencia tupla de información {T1 T2 S1 S2}, donde tupla se compone de la siguiente información

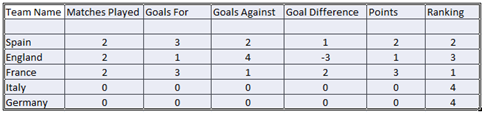
* T1 - Nombre del primer equipo
* T2 - Nombre del segundo equipo
* S1 - Goles anotados por el primer equipo
* S2 - Goles anotados por el segundo equipo

**Formato de salida: los**   
  
nombres de los equipos en orden de clasificación, un equipo por línea   
O   
Imprimir "Entrada inválida" cuando corresponda.

**Restricciones:**

1. **0 <N <= 10,000**
2. **0 <= S1, S2**

**Ejemplo:**   
  
Considera 5 equipos de España, Inglaterra, Francia, Italia y Alemania con los siguientes partidos:   
  
Partido 1: España vs. Inglaterra (3-0)   
(España obtiene 2 puntos, Inglaterra obtiene 0)   
Partido 2: Inglaterra vs. Francia (1 -1)   
(Inglaterra obtiene 1 punto, Francia obtiene 1)   
Partido 3: España vs. Francia (0-2)   
(España obtiene 0 puntos, Francia obtiene 2)

   
**Tabla 1. Tabla de puntos después de 3 coincidencias**

Dado que Italia y Alemania están empatados en puntos, la diferencia de goles se verifica. Ambos tienen lo mismo, entonces, Goals For está marcado. Como ambos son lo mismo Alemania e Italia comparten la 4 ªfila. Dado que Alemania aparece alfabéticamente antes de Italia, Alemania debe imprimirse antes de Italia.   
  
**Entonces el resultado final es:**  
Francia   
España   
Inglaterra   
Alemania   
Italia

**Muestra de entrada y salida**

|  |  |  |
| --- | --- | --- |
| **SNo.** | **Entrada** | **Salida** |
| 1 | 5  España Inglaterra Francia Italia Alemania  3  España Inglaterra 3 0  Inglaterra Francia 1 1  España Francia 0 2 | Francia  España  Inglaterra  Alemania  Italia |
| 2 | 5  España Inglaterra Francia Italia Alemania  3  España Inglaterra 3 0  Inglaterra Francia 1 1  España España 0 2 | entrada inválida |

***Nota*** :  
  
*Por favor, no use el paquete y el espacio de nombres en su código. Para lenguajes orientados a objetos, su código debe escribirse en una clase.*

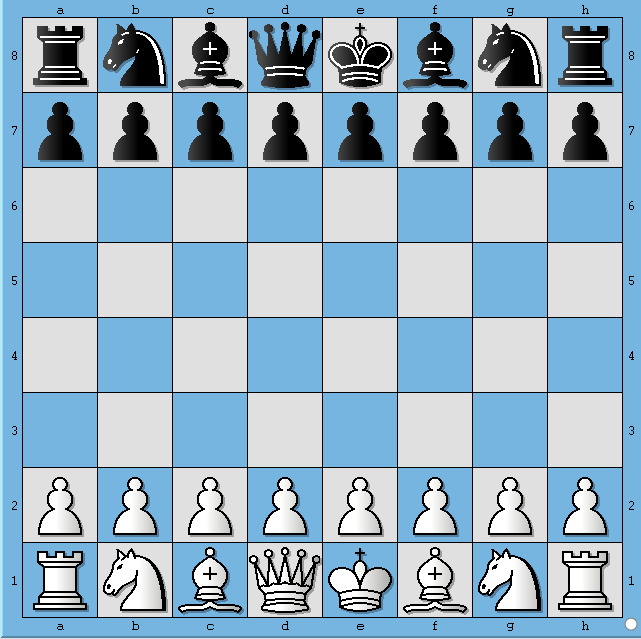
***Nota*** : Los  
  
*participantes que envían soluciones en lenguaje C no deben usar funciones de <conio.h> / <process.h>, ya que estos archivos no existen en gcc.*

***Nota*** :  
  
*Para C y C ++, el tipo de retorno de la función main () debe ser int.*

**© 2016 Tata Consultancy Services Limited. Todos los derechos reservados.**

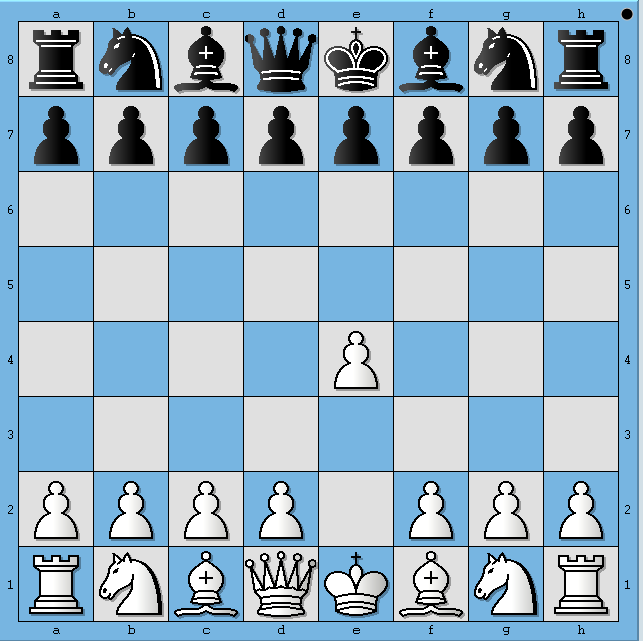
**Problema: Mate In One**

**Fondo**   
  
Una posición de tablero de ajedrez se captura con precisión mediante la notación Forsyth-Edwards y se abrevia como FEN. Un "registro" de FEN define una posición de juego particular, todo en una línea de texto y utilizando solo el juego de caracteres ASCII. Un registro FEN consta de seis campos. [Aquí](https://en.wikipedia.org/wiki/Forsyth%E2%80%93Edwards_Notation) puede encontrar una descripción completa del formato FEN para representar posiciones de Ajedrez.   
  
A los fines de este problema, solo considere el primero de los seis campos de FEN. Antes de describir el problema, veamos cómo FEN se mapea a una posición de tablero. Las siguientes 5 imágenes muestran las posiciones de la placa y su correspondiente representación FEN.

   
                    **Figura 1.**

|  |
| --- |
| Esta posición de la tabla representa la posición inicial antes de que cualquier lado haya realizado un movimiento. En formato FEN, esta posición de la placa se representa como    rnbqkbnr / pppppppp / 8/8/8/8 / PPPPPPPP / RNBQKBNR |

Digamos que White juega e4. Entonces la posición de la tabla se ve como se muestra a continuación 

   
                    **Figura 2.**

|  |
| --- |
| Esta posición de la tabla representa el tablero de ajedrez después de que las blancas hayan jugado e4. En formato FEN, esta posición de la placa se representa como    rnbqkbnr / pppppppp / 8/8 / 4P3 / 8 / PPPP1PPP / RNBQKBNR |

Del mismo modo, se representan 3 medios movimientos más en los siguientes diagramas

|  |  |  |
| --- | --- | --- |
| https://www.tcscodevita.com/CodevitaV6/images/cv15R2_R1P2_fig3.png | https://www.tcscodevita.com/CodevitaV6/images/cv15R2_R1P2_fig4.png | https://www.tcscodevita.com/CodevitaV6/images/cv15R2_R1P2_fig5.png |
| **Figura 3.** | **Figura 4.** | **Figura 5.** |

Los FEN correspondientes a las Figuras 3, 4 y 5 se representan como   
  
           3. rnbqkbnr / pppp1ppp / 8 / 4p3 / 4P3 / 8 / PPPP1PPP / RNBQKBNR   
           4. rnbqkbnr / pppp1ppp / 8 / 4p3 / 4PP2 / 8 / PPPP2PP / RNBQKBNR   
           5. rnbqkbnr / pppp1ppp / 8/8 / 4Pp2 / 8 / PPPP2PP / RNBQKBNR   
  
**Wikipedia describe el primer campo del formato FEN como sigue**   
  
Colocación de la pieza (desde la perspectiva de White). Cada rango se describe, comenzando con el rango 8 y terminando con el rango 1; dentro de cada rango, los contenidos de cada cuadrado se describen desde el archivo "a" hasta el archivo "h". Siguiendo la [notación algebraica estándar](https://en.wikipedia.org/wiki/Algebraic_chess_notation)(SAN), cada pieza se identifica con una sola letra tomada de los nombres estándar en inglés (peón = "P", caballero = "N", alfil = "B", torre = "R", reina = "Q" y rey = "K"). [1] Las piezas blancas se designan con letras mayúsculas ("PNBRQK"), mientras que las negras se usan en minúsculas ("pnbrqk"). Los recuadros vacíos se anotan usando los dígitos del 1 al 8 (el número de cuadrados vacíos) y "/" separa los niveles.   
  
**Declaración**   
  
Dada una posición del tablero en formato FEN, tu tarea es encontrar todos los movimientos que conducen a un mate forzado. Blanco para jugar y ganar en 1 movimiento.

**Formato de entrada:**

1. La primera línea contiene un solo registro FEN, que corresponde a una posición particular de la placa
2. La segunda línea contiene -1, lo que indica el final de la entrada

**Formato de salida:**

1. La salida debe imprimirse de la siguiente manera
   1. Una cadena "Mating moves" seguida de "[<move format>]"
   2. Donde <mover formato> se mueve representado en formato "[Pieza] [fromSquare] - [Pieza] [toSquare]"
   3. Si un movimiento de apareamiento implica una captura, represéntalo como "[Pieza] [desde la Plaza] - [Pieza] x [con la Plaza]"
   4. Si la Pieza que inflige una pareja es un peón, represéntala solo como [fromSquare] - [toSquare]
   5. Si hay más de un movimiento de apareamiento, entonces sigue las reglas dadas a continuación
      1. Los movimientos deben ordenarse alfabéticamente de acuerdo con sus valores ascii
      2. Dos movimientos deben estar separados por una coma seguida de un espacio en blanco
      3. Entre el último movimiento y el carácter "]" no debe haber espacio ni coma
2. Vea la sección Ejemplo para una mejor comprensión del formato de salida

**Restricciones:**

1. La posición del tablero siempre será blanca para moverse y aparearse en 1
2. Dado que nos centramos solo en la primera parte del FEN, esencialmente estamos ignorando la posibilidad de que Castling sea un movimiento de apareamiento. **Por lo tanto, nuestros casos de prueba no contienen FEN que den lugar a tales posiciones.**
3. No es necesario manejar posiciones de [*En Passant*](https://en.wikipedia.org/wiki/En_passant) . No hay casos de prueba que involucren movimientos *En Passant* .
4. No es necesario implementar reglas de [*promoción de peón*](https://en.wikipedia.org/wiki/Promotion_(chess)) . Nuestros casos de prueba no contienen posiciones que lleven a un peón a ser promovido e infligir un compañero.
5. No hay casos de prueba en los que la captura por un peón blanco conduce a un compañero.

**Muestra de entrada y salida**

|  |  |  |
| --- | --- | --- |
| **SNo.** | **Entrada** | **Salida** |
| 1 | 3k4 / 8 / 3p4 / 7B / 3Q4 / 2R1R3 / 8 / 5K2  -1 | Movimientos de apareamiento [Qd4-Qxd6] |
| 2 | 8 / R7 / 3k4 / 8 / 2Q1P1B1 / 8/8 / 2K1R3  -1 | Movimientos de apareamiento [Qc4-Qc7, Qc4-Qd5, e4-e5] |

**Explicación:**

|  |  |
| --- | --- |
| **Posición de la placa para la entrada de muestra 1: figura 6.**  https://www.tcscodevita.com/CodevitaV6/images/cv15ph2r1_R1P7_fig6.jpg | **Posición de la placa para la entrada de muestra 2: figura 7.**  https://www.tcscodevita.com/CodevitaV6/images/cv15ph2r1_R1P7_fig7.jpg |

***Nota*** :  
  
*Por favor, no use el paquete y el espacio de nombres en su código. Para lenguajes orientados a objetos, su código debe escribirse en una clase.*

***Nota*** : Los  
  
*participantes que envían soluciones en lenguaje C no deben usar funciones de <conio.h> / <process.h>, ya que estos archivos no existen en gcc.*

***Nota*** :  
  
*Para C y C ++, el tipo de retorno de la función main () debe ser int.*

**© 2016 Tata Consultancy Services Limited. Todos los derechos reservados.**