

CUADRADOS

DESCRIPCION

Hay una cuadrícula cuadrada llena de 0's y 1's. Dentro de ésta cuadrícula hay varios *subcuadrados* de diferentes tamaños que están formados exclusivamente por 0's. Por ejemplo en la siguiente cuadrícula

0	0	0	1
0	0	0	1
0	0	0	1
1	0	0	0

hay 1 *subcuadrado* de 3x3 formado únicamente por ceros, además hay 5 *subcuadrados* de 2x2 formados únicamente por ceros. Por último hay 12 *subcuadrados* de 1x1 que están formados por ceros.

PROBLEMA

Tu tarea consiste en escribir un programa, que dada una cuadrícula de $N \times N$ determine cuantos *subcuadrados* de ceros hay que sean de 1x1, 2x2, 3x3, ... , $N-1 \times N-1$, $N \times N$.

ENTRADA

Tu programa deberá leer del archivo de texto ENTRADA.TXT los siguientes datos.

En la primera línea el número $2 \leq N \leq 1,000$, que indica el tamaño de la cuadrícula. En las siguientes N líneas habrá N 0's o 1's separados por un espacio cada uno, que indican los valores de cada cuadro de la cuadrícula.

SALIDA

Tu programa deberá escribir al archivo de texto SALIDA.TXT N líneas. En cada línea deberá haber únicamente un número, que indique la cantidad de *subcuadrados* de tamaño $i \times i$ de 0's que hay en la cuadrícula.

Es decir, en la primera línea deberá escribir cuantos *subcuadrados* de 1x1 hay, en la segunda cuantos *subcuadrados* de 2x2, en la N -ésima línea deberá escribir cuantos *subcuadrados* de $N \times N$ hay.

EJEMPLO

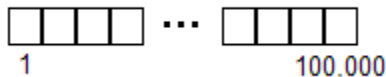
Los archivos de texto de entrada y salida para la cuadrícula que se muestra en la figura de la descripción son los siguientes

ENTRADA.TXT	SALIDA.TXT
4	12
0 0 0 1	5
0 0 0 1	1
0 0 0 1	0
1 0 0 0	

LA LINEA MAS LARGA

DESCRIPCIÓN

El juego de la línea más larga, es un juego para dos personas que se desarrolla de la siguiente manera, inicialmente se tiene un tablero vacío que puede tener hasta 100,000 casillas, estas casillas están ordenadas en forma de una larga línea.



Cada jugador tiene fichas de un color diferente, el jugador uno tiene fichas negras y el jugador dos tiene fichas azules. Durante su turno cada jugador puede colocar una de sus fichas en cualquiera de las casillas del tablero que este vacía.

El ganador del juego es aquél que después de un cierto número de turnos que se determinó al inicio del juego, haya formado la línea ininterrumpida más larga con sus fichas.

PROBLEMA

Eres el jugador número dos, y te encuentras en la última tirada del juego. Tu tarea consiste en escribir un programa que determine cual es la mejor jugada posible que puedes hacer.

La mejor jugada posible es aquella que forme la línea de fichas de tu color más larga.

ENTRADA

Tu programa deberá leer del archivo de texto ENTRADA.TXT los siguientes datos.

En la primera línea del archivo de entrada esta el número $4 \leq N \leq 100,000$ el cual indica el número de casillas con las que cuenta el tablero.

En la segunda línea habrá N números separados por un espacio. Estos números representan el estado del tablero al momento

de la última tirada. Cada uno de estos números puede tomar el valor 0, 1 ó 2, donde el número cero indica que la casilla en cuestión esta vacía. El número uno indica que la casilla tiene una ficha del jugador uno y el número dos indica una ficha del jugador dos, en este caso, una de tus fichas.

SALIDA

Tu programa deberá escribir en el archivo de texto SALIDA.TXT dos números, cada uno de ellos en una línea diferente.

En la primera línea deberá ir un número que indique la casilla en la que colocaste tu ficha. En la segunda línea deberá aparecer el largo de la línea que formaste con esa tirada.

EJEMPLO

Para ejemplo consideremos que hay un tablero de 16 casillas y que al momento de la última tirada la configuración del tablero es la siguiente



Como se explico es el último tiro y le corresponde al jugador dos (fichas azules). Del diagrama se puede observar que la mejor jugada que puede hacer el jugador azul es tirar su ficha en la posición 4, ya que con esa tirada forma una línea de longitud 3 con las fichas de las posiciones 4, 5 y 6.

Los archivos de entrada y salida para este ejemplo son:

ENTRADA . TXT	SALIDA . TXT
16	4
2 1 1 0 2 2 1 0 0 2 1 1 0 0 0 0	3

NOTA: Si existe más de una solución óptima, cualquiera de estas será considerada correcta.

LA MINA DE ACOSTA

DESCRIPCIÓN

Pachuca es una ciudad minera por tradición, de entre sus minas, una de las más famosas es la mina de Acosta. Al saber que venías a Pachuca al examen nacional de la OMI, tuviste curiosidad de conocer la famosa mina, sin embargo movido por la curiosidad y la fascinación te adentraste demasiado en el laberinto de la mina y ahora te encuentras perdido y aislado de tu grupo.

La mina es un laberinto formado por cuartos que se comunican unos con otros por medio de túneles, todos los túneles llevan a algún cuarto. Por último, la mina sólo tiene una salida.

PROBLEMA

Tu tarea consiste en desarrollar un programa que sea capaz de encontrar el camino más corto hacia la salida, a partir del cuarto en el que te encuentras en el momento en el que te diste cuenta que estabas perdido.

ENTRADA

Tu programa deberá leer del archivo de texto ENTRADA.TXT el mapa de la mina para determinar cual es el camino más corto a la salida.

El mapa de la mina está representado en el archivo de la siguiente manera.

En la primera línea del archivo hay un número $2 \leq N \leq 100$ que indica el número de cuartos que hay en la mina.

En las siguientes N líneas hay una lista de números separados por un espacio cada uno. Cada lista de números se refiere a uno de los cuartos de la mina, por ejemplo, la primera lista de números (línea 2 del archivo) se refiere al cuarto 1, la siguiente (línea 3 del archivo) al cuarto 2, etc.

El primer número de cada lista, es el número $0 \leq T_i \leq 100$, este número indica la cantidad de túneles que salen del cuarto i al número T_i le siguen T_i números en la lista, que indican los cuartos a los que conectan cada uno de los T_i túneles.

En la última línea del archivo hay dos números separados por un espacio, el primer número indica el cuarto en el que te encuentras, y el segundo número indica el cuarto que es la salida.

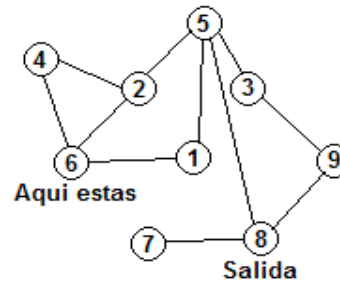
Los cuartos están numerados a partir del 1.

SALIDA

Tu programa deberá escribir en el archivo de texto SALIDA.TXT el camino más corto que haya desde el cuarto en el que te encuentras hasta el cuarto de salida.

Para describir el camino hacia la salida tu programa deberá escribir en una línea una lista de números separados por un espacio. Esta lista de números describe los cuartos por los que pasaste para llegar a la salida, comenzando por el cuarto en el que te encuentras hasta llegar a la salida.

EJEMPLO



Los archivos de entrada y salida para el mapa anterior quedan de la siguiente manera

ENTRADA . TXT	SALIDA.TXT
9	6 1 5 8
2 5 6	
3 4 5 6	
2 5 9	
2 2 6	
4 1 2 3 8	
3 1 2 4	
1 8	
3 5 7 9	
2 3 8	
6 8	

NOTA: Para los casos de prueba que se utilizarán, siempre habrá al menos un camino hacia la salida. En el caso de que exista más de un camino de la misma longitud, cualquiera de los caminos se considerará correcto, en el ejemplo anterior, la solución (6 2 5 8) también es correcta.

IMAGEN

DESCRIPCIÓN

Gracias a tus habilidades en el campo de la informática fuiste seleccionado por la Agencia Federal de Investigación (AFI). El área para la que fuiste contratado es la de procesamiento de imágenes. Tu tarea será trabajar en un nuevo algoritmo de compresión y codificación de imágenes. Después de analizar tu tarea te decides por los siguientes métodos, para la codificación un sistema de sustitución simple y para la compresión un sistema de codificación de largo de línea.

Los métodos que seleccionaste funcionan de la siguiente manera.

Codificación por sustitución simple: en éste método se cuenta con el conjunto de símbolos clave, cada símbolo representa una letra del alfabeto, un dígito o un espacio en blanco, por ejemplo una clave podría ser (%=O, \$=M, a=I), en éste caso el mensaje "%\$a" significa en realidad "OMI".

Compresión por largo de línea: éste tipo de compresión es muy útil para imágenes que tienen colores similares, en ésta compresión la imagen se va barriendo línea por línea y se forman grupos de puntos del mismo color. Supongamos que la siguiente figura es una imagen que queremos comprimir, para el ejemplo, sustituiremos cada punto de la imagen por una letra del alfabeto

```
AAAAAB
BBBCCC
CBAAAC
CCBBBB
```

la compresión queda

```
A5B5C5B1A3C4B5
```

indicando que la imagen tiene 5 puntos del color A, después 5 del B, 5 del C, 1 del B, etc. En la versión del método de compresión que estas usando **la imagen comienza siempre con un número que indica el ancho en puntos de la imagen.** Además en los casos en los que solo hay un punto de cierto color (como en el punto B en el tercer renglón de la imagen) no le sigue ningún número. Por último los grupos de puntos nunca son de más de 9 puntos, es decir, no puede haber un A25. En el caso de que hubiera 25 puntos del color A consecutivos, se tendría que poner A9A9A7. El archivo de la imagen comprimida completo queda

```
7A5B5C5BA3C4B5
```

El ancho en puntos de la imagen siempre es un número entre 2 y 500.

PROBLEMA

Debes escribir un programa que teniendo como entrada la clave de codificación y un archivo comprimido y codificado, sea capaz primero de decodificar el archivo y posteriormente de crear la imagen.

ENTRADA

Tu programa deberá leer del archivo de texto ENTRADA.TXT dos líneas.

La primera línea esta formada por 36 símbolos. Esta línea es la clave de codificación. Los primeros 26 símbolos representan las letras mayúsculas A..Z, sin contar la Ñ, CH, LL, ni la RR. Los últimos 10 símbolos representan los dígitos del 0..9. Los 36 símbolos están consecutivos y no tienen ningún espacio separándolos.

La segunda línea esta formada también por una serie de símbolos consecutivos que puede tener hasta 500 símbolos. La segunda línea representa el archivo de imagen comprimido y codificado.

SALIDA

En este problema, la salida consta de 2 partes, la primera es el archivo decodificado. Si logras decodificar el archivo correctamente obtendrás un 40% de los puntos. La segunda parte es crear la imagen, si reconstruyes la imagen correctamente obtendrás un 60% de los puntos del caso.

Tu programa deberá escribir en el archivo de texto SALIDA.TXT en la primera línea el archivo decodificado. En las siguientes líneas tu programa deberá escribir la imagen utilizando letras, las letras deberán ir sin ningún espacio entre ellas.

EJEMPLO

ENTRADA.TXT
A%ab&cd84)kegh792341L2xzv'm/'#ç-.:;! ;A-%-a-%A#aç%-

En este ejemplo el símbolo A representa a la letra A, el símbolo % representa a la letra B, etc.

SALIDA.TXT
7A5B5C5BA3C4B5 AAAAAB BBBCCC CBAAAC CCBBBB