

Problema 1: El Problema de los Caballos.

Hacer un programa que determine el numero máximo de caballos **M** ($1 \leq M \leq N^2$) que pueden ponerse en un tablero de ajedrez de **NXN** ($1 \leq N \leq 20$) sin atacarse. El programa debe desplegar al usuario todas las formas en que se pueden colocar los **M** caballos (máximo 10). Se debe justificar que el algoritmo cubre todas las posibles soluciones.

Entrada.

El número **N**

Salida.

Un archivo de texto de nombre CABAL_xx.TXT donde xx es el número **N** escrito con dos dígitos (01, 02, 03, ..., 19, 20). En la primera línea del archivo se encuentra el número **M**. La segunda línea contiene el número total de acomodos para los **M** caballos. A continuación deben estar las soluciones encontradas (máximo 10) codificadas de la siguiente manera:

Para cada casilla del tablero se escribe un 1 si hay un caballo en esa casilla, y un 0 en caso contrario. El tablero se representará por **N** líneas; en cada línea **N** dígitos, ya sean 1's o 0's, con un espacio entre cada par de dígitos.

Escriba una línea en blanco entre cada solución.

Ejemplo.

Entrada	CABAL_05.OUT
5	13 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

Problema 2: Tarjetas.

Se tiene una serie de n tarjetas; en cada tarjeta hay escrito un numero. Se quiere eliminar tarjetas de la serie, de tal forma que las restantes estén ordenadas de forma ascendente.

El programa debe escoger que tarjetas eliminar, para que el resultado tenga la mayor cantidad de tarjetas posibles.

Entrada.

La entrada es un archivo de texto “*.in”. En la primera línea se encuentra el número n ($1 \leq n \leq 1,000$) de tarjetas, y en las siguientes n líneas los números t_i ($0 \leq t_i \leq 200,000$) en cada tarjeta.

Salida.

La salida es un archivo con el mismo nombre que la entrada (si ya existe un archivo con ese nombre debe reemplazarlo), pero con la extensión “*.out”. En la primera línea esta k ($1 \leq k \leq n$) que es el número de tarjetas no eliminadas; en las siguientes k líneas están las tarjetas restantes, identificadas por su número de orden en el archivo de entrada.

Ejemplo.

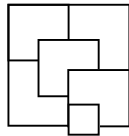
Prueba.in	Prueba.out
5	3
400	1
300	4
500	5
450	
480	

Problema 3: Acomodo de escuadras.

Se tiene un tablero de $2^n \times 2^n$ cuadros; el cual se quiere cubrir, a excepción de un cuadro, con piezas de la siguiente forma:



Escriba un programa que haga un recubrimiento para esas condiciones.



Entrada.

La entrada es un archivo de texto '*.IN'. En la primera línea se encuentra el número n ($0 \leq n \leq 10$). En la segunda se encuentran 2 números x, y ($1 \leq x \leq 2^n, 1 \leq y \leq 2^n$) que son las coordenadas del cuadro del tablero que debe quedar descubierto.

Salida.

En la primera línea el número m de escuadras utilizadas.

En las siguientes m líneas 3 valores: x, y, p , los cuales describen la posición de cada escuadra en el tablero.

(x, y) son las coordenadas del cuadro inferior izquierdo del cuadrado de 2×2 que contiene a la escuadra.

P representa la posición de la escuadra, puede tener uno de los siguientes valores dependiendo de la esquina que ocupe la escuadra en un cuadrado de 2×2 :

AI: Arriba a la izquierda.

AD: Arriba a la derecha.

BI: Abajo a la izquierda.

BD: Abajo a la derecha.

Algún despliegue de una solución posible.

Ejemplo.

EJEMPLO.IN	EJEMPLO.OUT
2	5
3, 1	1 3 AI
	3 3 AD
	1 1 BI
	2 2 AI
	3 1 AD