Ejercicio 1:

Diseñar e implementar un algoritmo utilizando la técnica de vuelta atrás para resolver este problema de forma óptima.

c) Razonar la Complejidad Temporal. ¿Cree que se podría resolver este problema de forma óptima utilizando algunas de las técnicas anteriores vistas durante el curso?

Sabemos que en el grado del árbol en cada nivel no disminuye al poder repetirse las posibilidades en cada dicho nivel. Con lo cuál ya sabemos que es una complejidad $O(m^n)$ donde "m" es el grado del árbol y "n" el número de niveles de dicho árbol (altura del árbol).

Sabemos que el grado del árbol es de 10 (10 posibilidades con las que rellenar una posición del tablero, desde el 0 hasta el 9 ambos incluidos).

El número de niveles (o altura del árbol) depende del número de "?" que haya en el tablero a rellenar, así que lo representaremos con la variable "n".

Concluimos que la complejidad temporal del algoritmo de backtracking para todas las soluciones y sin poda (lógicamente) en el problema del cuadrado numérico es $O(m^n) = O(10^n)$ siendo "n" el número de "?" a rellenar en el tablero.

Rellene la siguiente tabla:

TABLA (tiempos en milisegundos y CON_OPTIMIZACIÓN):
Pondremos "FdT" para tiempos > cinco minutos y "FdF" los < 50 msg

Caso de prueba	Tiempo para la primera solución	Tiempo para todas las soluciones	Número de soluciones encontradas
Test00	FdF	FdF	1
Test01	FdF	FdF	12
Test02	FdF	FdF	1
Test03	156	203	3
Test04	203	219	2
Test05	140	1031	5
Test06	FdF	1047	83
Test07	406	FdT	X