

PROBLEMA 1

MÁXIMO COMÚN DIVISOR Y MÍNIMO COMÚN MÚLTIPLO

NIVEL DE DIFICULTAD: BAJA

Escriba un programa que calcule el máximo común divisor y el mínimo común múltiplo de dos valores enteros. Se ha de escribir una función que determine el máximo común divisor y otra que calcule el mínimo común múltiplo

PROBLEMA 2

NÚMEROS ROMANOS

NIVEL DE DIFICULTAD: BAJA

Escriba un programa que convierta números enteros en números romanos. El programa debe disponer de una función que convierta de notación decimal a números romanos y otra que lo haga al contrario.

PROBLEMA 3

SISTEMA DE ECUACIONES LINEALES SIMULTÁNEAS

NIVEL DE DIFICULTAD: MODERADA

Escriba un programa que resuelve un sistema de ecuaciones lineales simultáneas utilizando el algoritmo de Gauss-Siedel. El programa debe ser capaz de:

- Leer los datos del sistema de ecuaciones ingresados por el usuario
- Determinar si el sistema de ecuaciones tiene solución
- En caso de haber solución, mostrarla por pantalla

PROBLEMA 4

EL MONTE

NIVEL DE DIFICULTAD: MODERADA

Un vector es un monte, si su primera mitad es creciente, la segunda decreciente, y el vector completo es un palíndromo. Diseñe una función que determine si un vector de N elementos (con $N > 0$) es un monto. Escriba el programa que demuestre la función.

PROBLEMA 5

EL NÚMERO PERFECTO

NIVEL DE DIFICULTAD: MODERADA

Se dice que un número N es perfecto si la suma de todos sus divisores, salvo él mismo, coincide con n . Por ejemplo, el número 6 es un número perfecto, ya que sus divisores son 1, 2, 3 y 6; sumando $1+2+3$ se obtiene 6. Implemente una función que indique si un número dado es perfecto. Escriba el programa que demuestra la función.

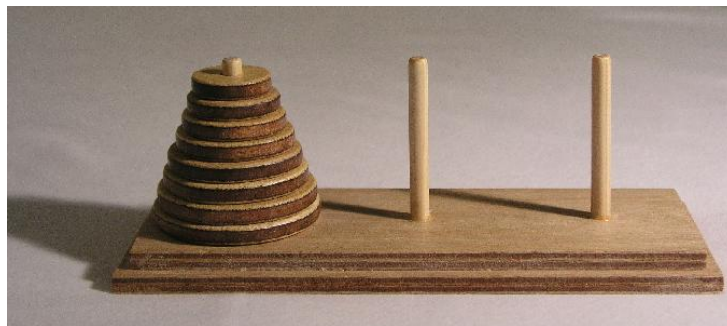
PROBLEMA 6

TORRES DE HANOI

NIVEL DE DIFICULTAD: MODERADA

Las Torres de Hanoi es un juego matemático clásico, que consiste en tres varillas verticales y un número indeterminado de discos que determinarán la complejidad de la solución. No hay dos discos iguales, están colocados de mayor a menor en la primera varilla ascendentemente, y no se puede colocar ningún disco mayor sobre uno menor a él en ningún momento. El juego consiste en pasar todos los discos a la tercera varilla colocados de mayor a menor ascendentemente.

Escriba un programa que dado un cierto número de discos K , desarrolle la solución del problema de las Torres de Hanoi.



PROBLEMA 7

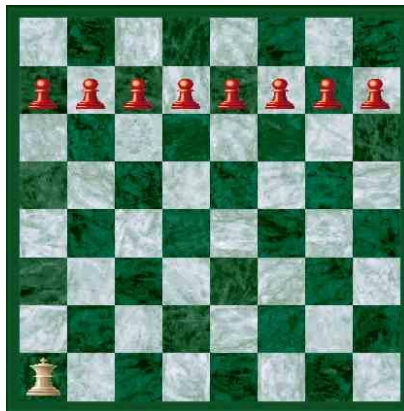
EL REY Y LOS 8 PEONES

NIVEL DE DIFICULTAD: MODERADA

El rey blanco y 8 peones negros se encuentran ubicados en un tablero de ajedrez. El rey y cada uno de los peones pueden "comerse" entre sí. El rey puede moverse de acuerdo a las reglas usuales del ajedrez, esto es, un casillero por vez, en cualquier dirección. Sin embargo, los peones no pueden moverse salvo para "comer" al rey; para ello, el rey debe ubicarse en cualquiera de los casilleros diagonales adyacentes al que ocupa el peón. Diremos que una "ruta" para el rey es **segura** si el rey puede recorrerla sin ser "comido" por ninguno de los 8 peones. Se trata entonces de encontrar un algoritmo que permita determinar si existe o no una ruta segura para el rey desde el casillero A1 al H8 para una posición dada de los 8 peones negros.

Escriba un programa que:

- Permita al usuario seleccionar las posiciones de los 8 peones negros
- Determine si existe o no una ruta segura para el rey, con respecto a la posición ingresada de los 8 peones negros, desde el casillero A1 al H8.
- Si existe tal ruta, que la describa



PROBLEMA 8

EL SALTO DEL CABALLO

NIVEL DE DIFICULTAD: MODERADA

El Salto del Caballo es un problema donde se pide determinar qué recorrido debe hacer un caballo en un tablero de ajedrez, para recorrer todas las casillas del tablero al menos una vez sin repetir ninguna casilla. El movimiento del caballo será de acuerdo a las reglas del juego de ajedrez. El programa debe:

- Permitir ingresar la posición inicial del caballo
- Determinar si existe solución
- Mostrar la solución encontrada

PROBLEMA 9

EL JUEGO DE LA VIDA

NIVEL DE DIFICULTAD: MODERADA-ALTA

Existe un asentamiento de K bacterias ubicadas en un grilla de M columnas por N filas, de modo que en cada celda hay a lo sumo una bacteria; por lo tanto, cada bacteria tiene como máximo 8 bacterias vecinas. El asentamiento está en continuo desarrollo debido a los nacimientos y defunciones que se producen. Las reglas de evolución son las siguientes:

- Aquellas bacterias que tienen 0, 1, 4, 5, 6, 7 u 8 vecinos mueren
- Las bacterias que tienen 2 o 3 vecinos sobreviven
- En cada celda vacía en cuya vecindad haya exactamente tres bacterias, nace una nueva bacteria
- Los insectos que nacen o mueren no afectan las reglas anteriores hasta que se haya completado el ciclo evolutivo.

Escriba un programa que modele la evolución del asentamiento y que:

- Permita ingresar las posiciones iniciales de las bacterias
- Muestre en pantalla el estado inicial del asentamiento, indicando las celdas ocupadas por bacterias
- Pueda mostrar el estado de evolución del asentamiento luego de L ciclos evolutivos

PROBLEMA 10

TIC-TAC-TOE (EL JUEGO DE LA VIEJA)

NIVEL DE DIFICULTAD: MODERADA-ALTA

El Tic-Tac-Toe consiste en un juego en el que en un tablero de 3 filas por 3 columnas, dos jugadores colocarán alternativamente X y O's. El ganador del juego es aquel que consiga primero colocar 3 X's o 3 O's alineados horizontal, vertical o diagonalmente. Escriba un programa que permita jugar Tic-Tac-Toe contra el computador

VERSIÓN INTELIGENTE

NIVEL DE DIFICULTAD: ALTA

La versión inteligente de este programa debe ser tal que su algoritmo básico de colocación de X's o O's sea aleatoria, pero que en la medida en que ejecute partidas, sea capaz de seleccionar jugadas en función de la probabilidad de éxito de cada una de ella. Se sugiere que el programa opere de la siguiente manera:

- El programa tendrá un registro de la probabilidad de éxito de cada posible jugada.
- Inicialmente, las probabilidades de éxito de cada jugada serán inicializadas en cero
- Cada vez que el programa gane, sumará 1 a la probabilidad de éxito de cada jugada hecha en la partida
- Cada vez que el programa pierda, restará 1 a la probabilidad de éxito de cada jugada hecha en la partida
- Cada vez que el programa deba decidir, revisará todas las jugadas disponibles a partir del estado actual del juego, y seleccionará aquella con mayor probabilidad de éxito
- Si todas las jugadas disponibles tienen igual probabilidad de éxito, seleccionará una aleatoriamente
- El programa debe almacenar el registro de probabilidades de éxito en disco, de tal manera que pueda recuperarlo la próxima vez que se ejecute.