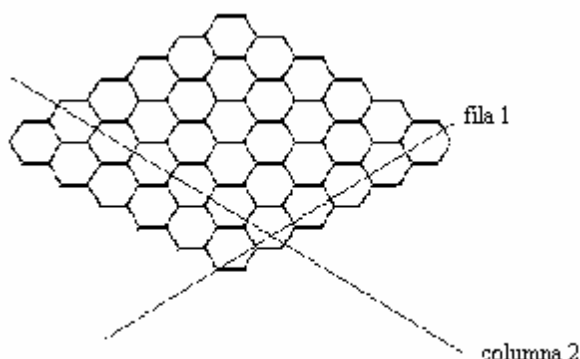


El Juego de Hex

El objetivo del juego para el primer jugador es conectar una ficha hex suya en la columna 1 a una ficha hex suya en la columna N.

Reglas de Hex:

Hex es un juego de estrategia jugado en un rombo de $N \times N$ hexágonos, como está ilustrado aquí para $N=6$.



1. Los dos jugadores del juego son su programa y la biblioteca de evaluación.
2. Su programa siempre hace la primera jugada.
3. Los jugadores colocan alternativamente fichas hex en el tablero.
4. Una ficha hex puede ser puesta en cualquier posición libre del tablero.
5. Dos hexágonos son adyacentes si comparten un lado.
6. Fichas hex en hexágonos adyacentes del mismo jugador (participante cerca a participante, o evaluador cerca a evaluador) están *conectadas*.
7. La conectividad es transitiva (y conmutativa): si el hex1 está conectado a hex2 y hex2 está conectado a hex3, entonces hex3 está conectado a hex1 y hex1 está conectado a hex3.

Tarea:

A usted se le pide escribir un programa que juegue el juego de Hex.

- El propósito del primer jugador (su programa) es conectar una ficha hex suya de la columna 1 a una ficha suya de la columna N.
- El otro jugador (el programa de evaluación) intenta conectar una ficha hex del evaluador en la fila 1 a una ficha del evaluador en la fila N.
- Si su programa juega optimamente, siempre ganará.

Entrada y Salida:

Su programa **no** debe leer o escribir en ningún archivo. Su programa **no** debe recibir entrada por teclado, y **no** debe producir salida en la pantalla. Recibirá toda su entrada de las funciones en la biblioteca hex. La biblioteca producirá un archivo de salida llamado HEX.OUT; usted debe ignorar su contenido.

Al comienzo del juego su programa será presentado con un tablero que puede tener fichas hex ya puestas, representando un estado de juego tal que el primer jugador puede aún ganar. Su programa

debe usar las funciones GetMax y LookAtBoard para determinar el estado del tablero.

Al comienzo del juego, un número igual de hexs pertenecen al programa de evaluación y a su programa.

Restricciones:

1. El tamaño del tablero siempre será un entero del rango de 1 a 20 inclusive.
2. Su programa puede hacer hasta 200 jugadas para completar un juego. El juego completo debe ser terminado dentro de 40 segundos. Se garantiza que la biblioteca de evaluación completará este proceso dentro de 20 segundos..

Biblioteca:

Se proporciona una biblioteca llamada *HexLib*, la cual usted debe vincular a su código. Un archivo ejemplo, para cada lenguaje de programación, mostrando como esto es hecho se incluye en el directorio de la tarea. Estos archivos son TESTHEX.CPP, TESTHEX.C, TESTHEX.PAS, y TESTHEX.BAS. Si usted está usando QuickBasic usted debe incluir la biblioteca escribiendo:

QB /L HEXLIB

Las funciones en HexLib son:

(en orden de Pascal, C/C++ y Basic respectivamente)

function LookAtBoard (row, column: integer): integer;
int LookAtBoard (int row, int column);
declare function LookAtBoard cdecl (byval x as integer, byval y as integer)

Devuelve

- 1 si fila<1 or fila>N o columna<1 o columna>N
- 0 si no hay una ficha hex en esa posición
- 1 si la ficha hex en la posición especificada pertenece a su programa (jugador 1)
- 2 si la ficha hex en la posición especificada pertenece al programa de evaluación (jugador 2)

procedure PutHex (row, column: integer);
void PutHex (int row, int column);
declare sub PutHex cdecl (byval x as integer, byval y as integer)
 Coloca una ficha hex del participante en la fila y columna especificadas si la posición no está ocupada.

function GameIsOver: integer;
int GameIsOver (void);
declare function GameIsOver cdecl ()

Devuelve uno de los siguientes enteros

- 0 el juego aún no ha terminado.
- 1 cada posición del tablero está ocupada por una ficha hex.
- 2 su programa ha ganado.
- 3 la biblioteca de evaluación ha ganado.

procedure MakeLibMove;
void MakeLibMove(void);
declare sub MakeLibMove cdecl ()

Permite que la biblioteca de evaluación calcular su siguiente movimiento y coloca su ficha hex en el tablero. El cambio al tablero será indicado por LookAtBoard y las otras funciones.

function GetRow: integer;
int GetRow (void);
declare function GetRow cdecl ()

Devuelve la fila en la cual la biblioteca de evaluación colocó una ficha, o -1 si no ha sido puesta todavía ninguna ficha. Esta función



siempre devuelve el mismo valor hasta que su programa llame MakeLibMove nuevamente.

function GetColumn: integer;

int GetColumn (void);

declare function GetColumn cdecl ()

Devuelve la columna en la cual la biblioteca de evaluación colocó una ficha, o -1 si no ha sido puesta todavía ninguna ficha. Esta función siempre devuelve el mismo valor hasta que su programa llame MakeLibMove nuevamente.

function GetMax: integer;

int GetMax (void);

declare function GetMax cdecl ()

Devuelve el tamaño del Tablero N.

Puntaje:

- Si su programa gana un juego, tendrá todos los puntos para ese conjunto de datos.
- Si su programa pierde un juego, obtendrá un 20% para ese conjunto de datos.
- Si su programa termina antes del fin de un juego o se le termina el tiempo, tendrá 0 puntos para ese conjunto de datos.