

## Ejercicio 121

**Enunciado: CUATRO EN RAYA.** Disponemos de un tablero vertical de dimensiones  $N \times N$ . Construya un algoritmo que dada una configuración del juego la evalúe, y devuelva la decisión que produce el resultado indicado.

**Resolución:** Las posiciones posibles que el jugador puede tomar son GANADORA, PERDEDORA y TABLAS. La decisión es la casilla que el jugador escoge para obtener la posición GANADORA o TABLAS. El tipo jugador en un int que contiene el tipo de ficha que él coloca. La generación de descendientes consiste en cualquier casilla del tablero que no este ocupada y que por debajo tenga sustento.

El caso base comprobamos que el otro jugador no ha ganado o el tablero esta lleno, para poder mover nosotros.

Interesa comentar el método:

```
int Victoria(int tablero[MAX][MAX],int Jug) .
```

Consta de cuatro partes que recorren todo el tablero buscando posibles secuencias de cuatro fichas del jugador que este jugando. Primero miramos horizontalmente, luego verticalmente, seguidamente miramos en las dos verticales.

### Ejemplo de la ejecución del programa:

```
Fila: 0  Columna: 0 1
Fila: 0  Columna: 1 2
Fila: 0  Columna: 2 1
Fila: 0  Columna: 3 2
Fila: 0  Columna: 4 0
Fila: 1  Columna: 0 2
Fila: 1  Columna: 1 1
Fila: 1  Columna: 2 1
Fila: 1  Columna: 3 2
Fila: 1  Columna: 4 0
Fila: 2  Columna: 0 2
Fila: 2  Columna: 1 1
Fila: 2  Columna: 2 2
Fila: 2  Columna: 3 1
Fila: 2  Columna: 4 0
Fila: 3  Columna: 0 1
Fila: 3  Columna: 1 2
Fila: 3  Columna: 2 1
Fila: 3  Columna: 3 1
```

```
Fila: 3  Columna: 4 0
Fila: 4  Columna: 0 1
Fila: 4  Columna: 1 1
Fila: 4  Columna: 2 1
Fila: 4  Columna: 3 2
Fila: 4  Columna: 4 2
1 2 1 2 0
2 1 1 2 0
2 1 2 1 0
1 2 1 1 0
1 1 1 2 2
GANADORA 3 4
```

```
Fila: 0  Columna: 0 1
Fila: 0  Columna: 1 2
Fila: 0  Columna: 2 1
Fila: 0  Columna: 3 2
Fila: 0  Columna: 4 0
Fila: 1  Columna: 0 2
Fila: 1  Columna: 1 1
Fila: 1  Columna: 2 1
Fila: 1  Columna: 3 2
Fila: 1  Columna: 4 0
Fila: 2  Columna: 0 2
Fila: 2  Columna: 1 1
Fila: 2  Columna: 2 2
Fila: 2  Columna: 3 2
Fila: 2  Columna: 4 0
Fila: 3  Columna: 0 1
Fila: 3  Columna: 1 2
Fila: 3  Columna: 2 2
Fila: 3  Columna: 3 1
Fila: 3  Columna: 4 0
Fila: 4  Columna: 0 1
Fila: 4  Columna: 1 2
Fila: 4  Columna: 2 1
Fila: 4  Columna: 3 1
Fila: 4  Columna: 4 1
1 2 1 2 0
2 1 1 2 0
2 1 2 2 0
1 2 2 1 0
1 2 1 1 1
PERDEDORA
```

## Código:

```
#include <stdio.h>
#define MAX 5

typedef enum{ GANADORA, PERDEDORA, TABLAS }TPosicion;
```

```

typedef struct
{
    int Fila;
    int Columna;
}TCasilla;

void Evaluar(int tablero[MAX][MAX],int JugA, int JugB, TPosicion *Pos,
             TCasilla *Dec);
void rellenar_tablero(int tablero[MAX][MAX]);
void imprimir(int tablero[MAX][MAX]);
int Valida(int Jug,int tablero[MAX][MAX],TCasilla aux);
int estaLleno(int tablero[MAX][MAX]);
int Victoria(int tablero[MAX][MAX],int Jug);

int main()
{
    int JugadorA,JugadorB;
    /*Tablero con tamaño fijo*/
    int tablero[MAX][MAX];
    TCasilla decision;
    TPosicion Pos;
    decision.Fila=0;
    decision.Columna=0;
    rellenar_tablero(tablero);
    imprimir(tablero);
    JugadorA= 1;
    JugadorB= 2;
    Evaluar(tablero, JugadorA, JugadorB, &Pos, &decision);
    if(Pos==GANADORA)
        printf("GANADORA %d %d \n", decision.Fila, decision.Columna);
    else{
        if(Pos==TABLAS)
            printf("TABLAS %d %d \n", decision.Fila, decision.Columna);
        else printf("PERDEDORA\n");
    }/* Fin else */
    return 0;
}/* Fin main */

void Evaluar(int tablero[MAX][MAX],int JugA, int JugB, TPosicion *Pos,
             TCasilla *Dec)
{
    int i,j;
    TPosicion PosDejamos;
    TCasilla DecX;
    TCasilla NewCasilla;
    if(Victoria(tablero,JugB) || estaLleno(tablero)){
        if(Victoria(tablero,JugB)) *Pos = PERDEDORA;
        else *Pos=TABLAS;
    }
    else{
        *Pos=PERDEDORA;
        for(i=MAX-1;i>=0 && Pos!=GANADORA;i--){
            for(j=0;j<MAX && Pos!=GANADORA;j++){
                NewCasilla.Fila=i;
                NewCasilla.Columna=j;
                if(Valida(JugA,tablero,NewCasilla)){
                    tablero[NewCasilla.Fila][NewCasilla.Columna]=JugA;
                    Evaluar(tablero,JugB,JugA,&PosDejamos,&DecX);
                    if(PosDejamos!=GANADORA){
                        *Dec=NewCasilla;
                        if(PosDejamos==TABLAS) *Pos=TABLAS;
                        else *Pos=GANADORA;
                    }/* Fin if */
                    tablero[NewCasilla.Fila][NewCasilla.Columna]=0;
                }/* Fin if */
            }/* Fin for */
        }/* Fin for */
    }
}

```

```
    /* Fin else */
/* Fin Evaluar */

void rellenar_tablero(int tablero[MAX][MAX])
{
    int i, j;
    for(i=0; i<MAX; i++){
        for(j=0; j<MAX; j++){ printf("Fila: %d  Columna: %d ", i, j);
            scanf("%d", &tablero[i][j]);
        }
    }
/* Fin rellenar_tablero */

void imprimir(int tablero[MAX][MAX])
{
    int i, j;
    for(i=0; i<MAX; i++){
        for(j=0; j<MAX; j++){ printf("%d ", tablero[i][j]);
            printf("\n");
        }
    }
/* Fin imprimir */

int Victoria(int tablero[MAX][MAX], int Jug)
{
    int i, j, salir=0, contador, contador1;
    /*Mirar horizontal*/
    contador=0;
    for(i=MAX-1; i>=0 && !salir; i--){
        for(j=0; j<MAX && !salir; j++){
            if(tablero[i][j]==Jug) contador++;
            else contador=0;
            if(contador==4) salir=1;
        }
        /* Fin if */
        contador=0;
    }
    /* Fin for */
    /*Mirar Vertical*/
    contador=0;
    for(i=0; i<MAX && !salir; i++){
        for(j=0; j<MAX && !salir; j++){
            if(tablero[j][i]==Jug) contador++;
            else contador=0;
            if(contador==4) salir=1;
        }
        /* Fin for */
        contador=0;
    }
    /* Fin for */
    /*Mirar Diagonal Izquierda*/
    contador=contador1=0;
    for(i=0; i<MAX && !salir; i++){
        for(j=0; j<=i && !salir; j++){
            if(tablero[j][i-j]==Jug) contador++;
            else contador=0;
            if(tablero[(MAX-1)-(i-j)][(MAX-1)-j]==Jug) contador1++;
            else contador1=0;
            if(contador==4 || contador1==4) salir=1;
        }
        /* Fin for */
        contador=contador1=0;
    }
    /* Fin for */
    /*Mirar Diagonal Derecha*/
    contador=contador1=0;
    for(i=0; i<MAX && !salir; i++){
        for(j=0; j<=i && !salir; j++){
            if(tablero[i-j][(MAX-1)-j]==Jug) contador++;
            else contador=0;
            if(tablero[(MAX-1)-j][i-j]==Jug) contador1++;
            else contador1=0;
            if(contador==4 || contador1==4) salir=1;
        }
        /* Fin for */
        contador=contador1=0;
    }
    /* Fin for */
}
```

```
        return salir;
    }/* Fin Victoria */

int Valida(int Jug,int tablero[MAX][MAX],TCasilla aux)
{
    int salir=0;
    if(aux.Fila<MAX && aux.Columna<MAX && aux.Fila>=0 && aux.Columna>=0){
        if(tablero[aux.Fila][aux.Columna]==0 &&
            tablero[aux.Fila+1][aux.Columna]!=0) salir=1;
    }/* Fin if */
    return salir;
}/* Fin Valida */

int estaLleno(int tablero[MAX][MAX])
{
    int i,j;
    int salir=1;
    for(i=0;i<MAX && salir;i++)
        for(j=0;j<MAX && salir;j++){ if(tablero[i][j]==0) salir=0;
    }/* Fin for */
    return salir;
}/* Fin estaLleno */
```