

Aunque C es uno de los mejores lenguajes de programación de propósito general, a medida que un sistema software se va haciendo más grande se va acentuando más algunas deficiencias del C, como es la casi ilimitada libertad que tiene el programador sobre las rutinas que se implementan. El C++ soluciona este problema facilitando la creación de unidades funcionales de caja negra que tienen acceso estrictamente controlado; a estas unidades se les llaman objetos, por este motivo se dice que el C++ es un lenguaje orientado a objetos. En el siguiente programa trabajaremos algunas funciones utilizadas tanto en C y C++. Algunos de estos son:

- Nuevo estilo de comentario (con `\\`).
- Entrada/Salida estándar (con los flujos `cout` y `cin`, operadores sobrecargados `<<` y `>>`, que se encuentran en el fichero de cabecera `iostream.h`).

Estos aplicados a un juego de memoria llamado concéntrese, para el cual se utilizó arrays bidimensionales para su estructuración. Un array es una colección de variables del mismo tipo que se referencia por un nombre común. A un elemento específico de un array se accede mediante un índice. En C todos los arrays constan de posiciones de memoria contiguas. La dirección más baja corresponde al primer elemento y la dirección más alta al último elemento. Los arrays pueden tener de una a varias dimensiones. Un array bidimensional es, en realidad, un array unidimensional donde cada elemento es otro array unidimensional. Los arrays bidimensionales son un caso particular de los arrays multidimensionales. Así como a los arrays unidimensionales se les suele llamar vectores, a los arrays bidimensionales se les suele llamar matrices.

En el programa manejamos diferentes tipos de matrices:

```
int matRes[10][10], //Matriz tipo entero de respuestas.
int mR[10][10];    //Matriz tipo entero de respuestas.
char matPreg[10][10]; //Matriz tipo char donde cargamos las letras o símbolos.
```

Las cuales utilizamos por medio de las siguientes funciones:

```
void limpiarPreg //Coloca en NULL las posiciones de la matriz.
void limpiarRes //Coloca en cero las posiciones de la matriz auxiliar.
void cargarMat   //Cargamos la matriz con los símbolos.
void mostrarPreg //Muestra la matriz de símbolos.
void mostrarPreg2 //Muestra solo el símbolo de una posición dada de la matriz.
void mostrarRes  //Muestra la matriz de respuesta.
void jugarConcentre //Realizamos las jugadas.
char verificar    //Verificamos y buscamos el elemento en la matriz para saber
                  //las posiciones de un número.
void intercambiar //Recuperamos lo que tenían esas posiciones de la matriz.
```

El programa concéntrese, es el típico juego de buscar parejas en un tablero formado por secuencias de números o letras organizados en una matriz. Este juego también es llamado juego de Memoria, Bloques de Memoria o Parejas.

Para comenzar el juego, se elige el tamaño del tablero, por ejemplo: Si el tamaño es 8 serían 8 columnas por 8 filas dando 64 casillas correspondientes, en las cuales habría 32 parejas de letras o símbolos (fig.1), las cuales se visualizarían unos segundos según el tamaño del tablero. Luego hay que elegir un primer número, seguido de la tecla Enter, y en el tablero se mostrara la letra o símbolo oculto detrás de ese número, luego elegimos el segundo número seguido de la misma tecla y observaremos en el tablero la letra o símbolo oculto detrás de ese segundo número.

Visualizaremos por unos segundos las 2 parejas de letras o símbolos. Si son iguales, los números correspondientes desaparecerán del tablero dejando en su lugar el siguiente símbolo [] y se le dará un punto al jugador actual, el cual seguirá jugando. Si son diferentes el turno será para el siguiente jugador. El juego termina cuando el número de parejas ocultas se hallan encontrado o si el jugador digita cero (0), cuando se le pregunte si desea salir del juego. Los puntajes de los jugadores se visualizarán constantemente en pantalla, y el ganador será el que tenga mayor puntaje, el cual se mostrara cuando el juego finalice.

Fig.1

Ejemplo:

Digite el Tamaño del Tablero: 8

MATRIZ DE NÚMEROS

	0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7	8
1	9	10	11	12	13	14	15	16
2	17	18	19	20	21	22	23	24
3	25	26	27	28	29	30	31	32
4	33	34	35	36	37	38	39	40
5	41	42	43	44	45	46	47	48
6	49	50	51	52	53	54	55	56
7	57	58	59	60	61	62	63	64

MATRIZ DE LETRAS O SÍMBOLOS

	0	1	2	3	4	5	6	7
0	A	O	N	M	W	B	P	O
1	V	C	E	L	^	X	L	-
2	T	I	J	X	Y	U	D	G
3	/	J	A	V	'	B	U	P
4	R	Q	J	J	Y	/	G	Q
5	S	C	T	E	W	F	[K
6	-	S	Z	D	F	Z	K	'
7	R	I	N	M	H	^	[H

JUEGO CONCENTRECE

Puntaje del Jugador 1: 0

Puntaje del Jugador 2: 0

Jugador 1

Escoja los números: (0 para salir del juego):

Si el jugador 1 escoge los números 1 y 12

1 = A (i,j)=(0,0)
 12 = L (i,j)=(1,3) A!=L "NO SON PAREJAS"

Si el jugador 1 escoge los números 1 y 27

1 = A (i,j)=(0,0)
 27 = A (i,j)=(3,2) A=A "SON PAREJAS"

DATOS DEL PROGRAMA

Nombre del programa: Concéntrase

Nombre del archivo fuente: CONCEN.CPP

Nombre del archivo ejecutable: CONCEN.EXE

Fecha de creación: 09-03-2005

Compilador usado: TURBO C++ IDE 3.0

ALGORITMO

La estructura del programa en C o C++ quedaría así:

[directivas del pre-procesador: includes y defines]
[declaración de variables globales]
[prototipos de funciones]
[función main]
[definiciones de funciones]

[declaración de variables globales]:

```
int matRes[10][10], mR[10][10]; //Matriz de respuestas.
char matPreg[10][10]; //Cargados las letras o símbolos
int i, j, tam;
int posX=0, posY=0; //Posiciones que encontramos
int posX[2], posY[2]; //Guardamos las posiciones encontradas
int px=0, py=0; //Índice de los vectores
```

[función main]:

```
main()
{
    Repetir
    {
        Escribir(Digite el Tamaño del Tablero:);
        lea(tam); //Tamaño de la matriz Para jugar

        Si( ((tam*tam)%2 == 0) && (tam >= 4) )
        {
            limpiarPreg(matPreg, tam); //Coloca en NULL las posiciones de la matriz
            cargarMat(matPreg, tam); //Llenamos la matriz con los símbolos
            mostrarPreg(matPreg, tam, NULL); //Mostramos la matriz
            pausa(tam*500); //pausa.
            limpiarRes(matRes, tam); //Ponemos en cero las posiciones de la matriz de resultados
            limpiarRes(mR, tam); //Ponemos en cero la matriz auxiliar
            jugarConcentrece(matPreg, matRes, mR, tam); //jugamos
        }
        Finsi

        Sino
        {
            Escribir(El tamaño debe ser mínimo de [ 4 ] y par);
        }

        Hasta (tam < 4) FinRepetir
    }
    Fin
}
```

[definiciones de funciones]:

```
//*****
//Ponemos en NULL la matriz de símbolos
limpiarPreg(char matPreg[10][10], int tam)
{
    Inicio
    {
        Para(i=0; i<tam; i++)
        {
            Para(j=0; j<tam; j++)
            {
                matPreg[i][j] = NULL; //Ponemos en NULL
            }
        }
    }
    FinPara
}

//*****
//Ponemos en cero matriz de respuesta.
limpiarRes(int matRes[10][10], int tam)
{
    Inicio
    {
        Para(i=0; i<tam; i++)
        {
            Para(j=0; j<tam; j++)
            {
                matRes[i][j] = 0; //Ponemos en 0
            }
        }
    }
    FinPara
}

Fin
```

```

//*****
//Cargamos la matriz con los símbolos de forma aleatoria.
cargarMat(char matPreg[10][10], int tam)
Inicio
    int a, b;

    Función Aleatoria();
    Para(a=1; a<=(tam*tam)/2; a++) //n-pares depende el tamaño.
        Para(b=0; b<2; ) //Solo dos veces que son las parejas
            i = Aleatorio(tam); //Busca una posición cualquiera Para la fila.
            j = Aleatorio(tam); //Busca una posición cualquiera Para la columna.

            Si(matPreg[i][j] == NULL) //Si es diferente de NULL ingresa
                matPreg[i][j] = a+64; //65 - A
                b++; //Incrementamos Si ingresa.

            FinSi
        FinPara
    FinPara
Fin

//*****
//Mostramos la matriz de símbolos
mostrarPreg(char matPreg[10][10], int tam, char c)
Inicio
    int x=5, y=5;

    Para(i=0; i<tam; i++)
        x = 5;

        Para(j=0; j<tam; j++)
            Si(matPreg[i][j] != c)
                ir a xy(x,y);
                Escribir(matPreg[i][j]);

            FinSi
            x += 5;

        FinPara
        y += 1;
    FinPara
Fin

//*****
//Mostramos solo el símbolo de una posición dada.
mostrarPreg2(char matPreg[10][10], char num, int tam,int posx, int posy)
Inicio
    int x=5, y=5;
    Para(i=0; i<tam; i++)
        x = 5;

        Para(j=0; j<tam; j++)
            Si(matPreg[i][j] == num) //Si es igual al dato mandado
                Si( (posx == i) && (posy == j) ) //y si esta en las posiciones dadas
                    ir a xy(x,y);
                    Escribir(" ");
                    ir a xy(x,y);
                    Escribir(matPreg[i][j]);
                    i=tam;
                    j=tam;

                FinSi
            FinSi
            x += 5;

        FinPara
        y += 1;
    FinPara
Fin

```

```

//*****
//Mostramos la matriz de respuesta
mostrarRes(int matRes[10][10], int tam, int dSi)
Inicio
    int x=5, y=5;

    Para(i=0; i<tam; i++)
        x = 5;

        Para(j=0; j<tam; j++)

            Si(matRes[i][j] != dSi) //Si es diferente de lo que mandamos
                ir a xy(x,y);
                Escribir(matRes[i][j]);

            FinSi

            Sino //Sino imprime [] Para saber que son iguales
                ir a xy(x,y);
                Escribir("[ ]");
            FinSino
            x += 5;

        FinPara
        y += 1;

    FinPara

Fin

//*****
//Jugamos
jugarConcentrece(char matPreg[10][10], int matRes[10][10], int mR[10][10], int tam)
Inicio
    char num1, num2; //Donde almacenamos las figuras escogidas
    int n=1, sw=0, jug=0;
    int op; //Numero de la matriz respuesta
    int jugador=1, jd=1; //Jugadores
    int puntjug1=0, puntjug2=0; //Puntaje de los jugadores
    int cent=0; //Controla el Fin del juego.

    Para(i=0; i<tam; i++) //Cargan los números en la matriz 1, 2, 3,...

        Para(j=0; j<tam; j++)
            matRes[i][j] = n; //Cargamos
            n++;
        FinPara

    FinPara

    Repetir
        Si(sw == 0) //Si es la primera vez
            limpiarpantalla
            mostrarRes(matRes, tam, 0);
            sw = 1;
        FinSi

        ir a xy(10,14); Escribir("JUEGO CONCÉNTRERE ");
        ir a xy(10,15); Escribir("Puntaje del Jugador 1: ",puntjug1);
        ir a xy(10,16); Escribir("Puntaje del Jugador 2: ",puntjug2);
        ir a xy(10,18); Escribir("Jugador ",jd);
        ir a xy(10,19); Escribir("Escoja los números: (0 Para Salir del Juego):");
        ir a xy(48,20); lea(op);

        Si(op==0){
            Escribir("FINALIZANDO JUEGO CONCÉNTRERE ");
            pausa(4000);
            salir;
        }
        FinSi

```

```

Si(sw == 1)
    //Ingresamos lo que devuelve verificar.
    num1 = verificar(matPreg, matRes, mR, op, tam);
    sw = 2; //Pasamos a la 2da jugada
    jug++; //Incrementamos la jugada
    //Mostramos lo que sé a jugado
    mostrarPreg2(matPreg, num1, tam, posx[0], posy[0]);
FinSi

Sino Si(sw == 2)
    num2 = verificar(matPreg, matRes, mR, op, tam);
    sw = 1; //Pasamos a la 1ra jugada.
    jug++;
    mostrarPreg2(matPreg, num2, tam, posx[1], posy[1]);
FinSino

Si(num1 == num2) //Si son iguales
    pausa(2000);

    Si(jugador == 1)
        puntjug1 += 1; //Incrementamos el puntaje del Jug 1
    FinSi

    Sino Si(jugador == 2)
        puntjug2 += 1; //Incrementamos el puntaje del Jug 2
    FinSino

    //Eliminamos si son iguales
    matRes[posx[0]][posy[0]] = 0;
    matRes[posx[1]][posy[1]] = 0;
    sw=0; //Ponemos a mostrar como al principio.
    jug=0; //Limpiamos las jugadas
    px=0; //Limpiamos índices de las posiciones del vector posx
    py=0; //Limpiamos índices de las posiciones del vector posy

FinSi

Si(jug > 1) //Si no son iguales
    pausa(2000);
    jug=0; //Limpiamos las jugadas
    //Recuperamos lo que tenían esas posiciones
    intercambiar(mR, matRes, posx[0], posy[0]);
    intercambiar(mR, matRes, posx[1], posy[1]);
    sw=0;
    px=0;
    py=0;
    num1=NULL;
    num2=NULL;
    //Si se pierde
    Si(jugador == 1) //Si es el jugador 1
        jugador=2; //Pasamos al jugador 2
        jd=2;
    FinSi

    Sino
        Si(jugador == 2) //Si es el jugador 2
            jugador=1; //Pasamos al jugador 1
            jd=1;
        FinSi
    FinSino

Para(int p=0; p<tam; p++) //Verificamos si no hay mas jugadas en la matriz
    Para(int q=0; q<tam; q++)
        Si(matRes[p][q] != 0)
            q=tam;
            p=tam;
            cent=0; //Si hay jugadas
        FinSi

```

```

        Sino
            cent=1; //Si no hay jugadas
        FinPara

    FinPara

hasta (cent != 1) FinRepetir;

Si(puntjug1 > puntjug2) //Si el ganador es el jugador 1
    ir a xy(10,22);
    Escribir("El Ganador es el Jugador 1 con un puntaje de: ",puntjug1);
FinSi
Si(puntjug2 > puntjug1) //Si el ganador es el jugador 2
    ir a xy(10,22);
    Escribir("El Ganador es el Jugador 2 con un puntaje de: ",puntjug2);
FinSi
Si(puntjug2 == puntjug1) //si hay empate
    ir a xy(10,22);
    Escribir("Juego Empatado con puntajes de:",puntjug1);
FinSi

Fin

//*****
//verificamos y buscamos el elemento Para saber las posiciones de un numero
//dada
char verificar(char matPreg[10][10], int matRes[10][10], int mR[10][10],int num, int tam)
Inicio
    int x, y;
    char aux;
    int posX=0, posY=0;

    Para(x=0; x<tam; x++)
        Para(y=0; y<tam; y++)
            Si(num == matRes[x][y])
                posX = x;
                posY = y;
                posX[px] = posX; //Guardamos la posición X
                posY[py] = posY; //Guardamos la posición Y
                x = tam; //
                y = tam; // Bloqueamos
                px++; //
                py++; // Incrementamos posiciones del vector
            FinSi
        FinPara
    FinPara
    aux = matPreg[posX][posY]; //Almacenamos el elemento que encontramos en la
                                //posición que resulto.
    intercambiar(matRes, mR, posX, posY); //Copiamos el elemento de la
                                //matriz resultado a la matriz auxiliar

    retornar aux; //Retornamos el elemento que encontramos
Fin

//*****
//Copiamos.
void intercambiar(int matRes[10][10], int mR[10][10], int posX, int posY)
Inicio
    char aux;
    mR[posX][posY] = matRes[posX][posY];
Fin

```

CÓDIGO FUENTE

```
#include <iostream.h> //cout, cin
#include <conio.h> //getch(), clrscr()
#include <stdlib.h> //random y randomize
#include <dos.h> //delay()

void limpiarPreg(char matPreg[10][10], int tam);
void limpiarRes(int matRes[10][10], int tam);
void cargarMat(char matPreg[10][10], int tam);
void mostrarPreg(char matPreg[10][10], int tam, char c);
void mostrarPreg2(char matPreg[10][10], char num, int tam, int posX, int posY);
void mostrarRes(int matRes[10][10], int tam, int dif);
void jugarConcentrece(char matPreg[10][10], int matRes[10][10], int mR[10][10], int tam);
char verificar(char matPreg[10][10], int matRes[10][10], int mR[10][10], int num, int tam);
void mostrarRes(char matPreg[10][10], int tam, int posX, int posY);
void intercambiar(int matRes[10][10], int matPreg[10][10], int posX, int posY);

int matRes[10][10], mR[10][10]; //Matriz de respuestas.
char matPreg[10][10]; //Cargados las letras o símbolos
int i, j, tam;
int posX=0, posY=0; //Posiciones que encontramos
int posX[2], posY[2]; //Guardamos las posiciones encontradas
int px=0, py=0; //Indicie de los vectores

void main()
{
    int tam;
    do {
        clrscr();
        cout<<"Digite el Tamaño del Tablero: ";
        cin>>tam; //Tamaño de la matriz para jugar
        if( ((tam*tam)%2 == 0) && (tam >= 4) )
        {
            clrscr();
            limpiarPreg(matPreg, tam); //Pone en NULL las posiciones de la matriz
            cargarMat(matPreg, tam); //Llenamos la matriz con los símbolos
            mostrarPreg(matPreg, tam, NULL); //Mostramos la matriz
            delay(tam*500); //pausa.

            limpiarRes(matRes, tam); //Ponemos en cero las posiciones de la matriz de resultados
            limpiarRes(mR, tam); //Ponemos en cero la matriz auxiliar
            jugarConcentrece(matPreg, matRes, mR, tam); //Jugamos
        }
        else
            cout<<"\nEl tamaño debe ser mínimo de [ 4 ] y par\n";
        getch();
    }while(tam < 4);
}
//Ponemos en NULL la matriz de símbolos
void limpiarPreg(char matPreg[10][10], int tam)
{
    for(i=0; i<tam; i++)
    {
        for(j=0; j<tam; j++)
            matPreg[i][j] = NULL; //Ponemos en NULL
    }
}
//Ponemos en cero matriz de respuesta.
void limpiarRes(int matRes[10][10], int tam)
{
    for(i=0; i<tam; i++)
    {
        for(j=0; j<tam; j++)
            matRes[i][j] = 0; //Ponemos en 0
    }
}
```



```

//Cargamos la matriz con los símbolos.
void cargarMat(char matPreg[10][10], int tam)
{
    int a, b;

    randomize();
    for(a=1; a<=(tam*tam)/2; a++) //n-pares depende el tamaño.
    {
        for(b=0; b<2; ) //Solo dos veces que son las parejas
        {
            i = random(tam); //Busca una posición cualquiera para la fila.
            j = random(tam); //Busca una posición cualquiera para la columna.

            if(matPreg[i][j] == NULL) //Si es diferente de NULL ingresa
            {
                matPreg[i][j] = a+64; //65 - A
                b++; //Incrementamos si ingresa.
            }
        }
    }
}

//Mostramos la matriz de símbolos
void mostrarPreg(char matPreg[10][10], int tam, char c)
{
    int x=5, y=5;
    for(i=0; i<tam; i++)
    {
        x = 5;
        for(j=0; j<tam; j++)
        {
            if(matPreg[i][j] != c)
            {
                gotoxy(x,y);
                cout<<matPreg[i][j];
            }
            x += 5;
        }
        y += 1;
    }
}

//Mostramos solo el símbolo de una posición dada.
void mostrarPreg2(char matPreg[10][10], char num, int tam,
                  int posx, int posy)
{
    int x=5, y=5;
    for(i=0; i<tam; i++)
    {
        x = 5;
        for(j=0; j<tam; j++)
        {
            if(matPreg[i][j] == num) //Si es igual al dato mandado
            {
                if( (posx == i) && (posy == j) ) //y si esta en las posiciones dadas
                {
                    gotoxy(x,y);
                    cout<<" ";
                    gotoxy(x,y);
                    cout<<matPreg[i][j];
                    i=tam;
                    j=tam;
                }
            }
            x += 5;
        }
        y += 1;
    }
}

```

```

//Mostramos la matriz de respuesta
void mostrarRes(int matRes[10][10], int tam, int dif)
{
    int x=5, y=5;
    for(i=0; i<tam; i++)
    {
        x = 5;
        for(j=0; j<tam; j++)
        {
            if(matRes[i][j] != dif) //Si es diferente de lo que mandamos
                                   //cuando no son iguales.
            {
                gotoxy(x,y);
                cout<<matRes[i][j];
            }
            else { //Sino imprime [] para saber que son iguales
                gotoxy(x,y);
                cout<<"[]";
            }
            x += 5;
        }
        y += 1;
    }
}

//Jugamos
void jugarConcentrece(char matPreg[10][10], int matRes[10][10],int mR[10][10],int tam)
{
    char num1, num2; //Donde almacenamos las figuras escogidas
    int n=1, sw=0, jug=0;
    int op; //Numero de la matriz respuesta
    int jugador=1, jd=1; //Jugadores
    int puntjug1=0, puntjug2=0; //Puntaje de los jugadores
    int cent=0; //Controla el fin del juego.

    for(i=0; i<tam; i++) //Cargan los números en la matriz 1, 2, 3,...
    {
        for(j=0; j<tam; j++)
        {
            matRes[i][j] = n; //Cargamos
            n++;
        }
    }
    do {
        if(sw == 0) //Si es la primera vez
        {
            clrscr();
            mostrarRes(matRes, tam, 0);
            sw = 1;
        }
        gotoxy(10,14); cout<<"\nJUEGO CONCÉNTRERE ";
        gotoxy(10,15); cout<<"\nPuntaje del Jugador 1: "<<puntjug1;
        gotoxy(10,16); cout<<"\nPuntaje del Jugador 2: "<<puntjug2;
        gotoxy(10,18); cout<<"\nJugador "<<jd;
        gotoxy(10,19); cout<<"\nEscoja los numeros: (0 para Salir del Juego):";
        gotoxy(48,20); cout<<" ";
        gotoxy(48,20); cin>>op;
        if(op==0){
            cout<<"\n FINALIZANDO JUEGO CONCÉNTRERE ";
            delay(4000);
            exit(0);
        }

        if(sw == 1)
        {
            //Ingresamos lo que devuelve verificar.
            num1 = verificar (matPreg, matRes, mR, op, tam);
            sw = 2; //Pasamos a la 2da jugada
            jug++; //Incrementamos la jugada
            //Mostramos lo que se a jugado
            mostrarPreg2(matPreg, num1, tam, posX[0], posY[0]);
        }
    }
}

```

```

else if(sw == 2)
{
    num2 = verificar (matPreg, matRes, mR, op, tam);
    sw = 1; //Pasamos a la 1ra jugada.
    jug++;
    mostrarPreg2(matPreg, num2, tam, posx[1], posy[1]);
}

if(num1 == num2) //Si son iguales
{
    delay(2000);
    if(jugador == 1)
    {
        puntjug1 += 1; //Incrementamos el puntaje del Jug 1
    }
    else if(jugador == 2)
    {
        puntjug2 += 1; //Incrementamos el puntaje del Jug 2
    }
    //eliminamos si son iguales
    matRes[posx[0]][posy[0]] = 0;
    matRes[posx[1]][posy[1]] = 0;
    sw=0; //Ponemos a mostrar como al principio.
    jug=0; //Limpiamos las jugadas
    px=0; //Limpiamos índices de las posiciones del vector posx
    py=0; //Limpiamos índices de las posiciones del vector posy
}
if(jug > 1) //Si no son iguales
{
    delay(2000);
    jug=0; //Limpiamos las jugadas
    //Recuperamos lo que tenían esas posiciones
    intercambiar(mR, matRes, posx[0], posy[0]);
    intercambiar(mR, matRes, posx[1], posy[1]);
    sw=0;
    px=0;
    py=0;
    num1=NULL;
    num2=NULL;
    //Si se pierde
    if(jugador == 1) //Si es el jugador 1
    {
        jugador=2; //Pasamos al jugador 2
        jd=2;
    }
    else
    {
        if(jugador == 2) //Si es el jugador 2
        {
            jugador=1; //Pasamos al jugador 1
            jd=1;
        }
    }
}

//Verificamos si no hay mas jugadas en la matriz
for(int p=0; p<tam; p++)
{
    for(int q=0; q<tam; q++)
    {
        if(matRes[p][q] != 0)
        {
            q=tam;
            p=tam;
            cent=0; //Si hay jugadas
        }
        else
            cent=1; //Si no hay jugadas
    }
}
//
}while(cent != 1);

```

```

    if(puntjug1 > puntjug2) //Si el ganador es el jug 1
    {
        gotoxy(10,22);
        cout<<"El Ganador es él << Jugador 1 >> con un puntaje de: "<<puntjug1;
    }
    if(puntjug2 > puntjug1)
    {
        //Si el ganador es el Jug 2
        gotoxy(10,22);
        cout<<"El Ganador es él << Jugador 2 >> con un puntaje de: "<<puntjug2;
    }
    if(puntjug1 == puntjug2) //si hay empate
    {
        gotoxy(10,22);
        cout<<"Juego Empatado con puntajes de:"<<puntjug1;
    }
}

```

//Verificamos y buscamos el elemento para saber las posiciones de un numero
//dada

```

char verificar(char matPreg[10][10], int matRes[10][10], int mR[10][10],
               int num, int tam)
{
    int x, y;
    char aux;
    int posX=0, posY=0;

    for(x=0; x<tam; x++)
    {
        for(y=0; y<tam; y++)
        {
            if(num == matRes[x][y])
            {
                posX = x;
                posY = y;
                posX[px] = posX; //guardamos la posición x
                posY[py] = posY; //guardamos la posición y
                x = tam; //
                y = tam; // Bloqueamos
                px++; //
                py++; // incrementamos posiciones del vector
            }
        }
    }
    aux = matPreg[posX][posY]; //Almacenamos el elemento que encontramos en la
                               //posición que resulto.
    intercambiar(matRes, mR, posX, posY); //Copiamos el elemento de la
                                           //matriz resultado a la matriz
                                           //auxiliar
    return aux; //Retornamos el elemento que encontramos
}

```

//Copiamos.

```

void intercambiar(int matRes[10][10], int mR[10][10], int posX, int posY)
{
    char aux;
    mR[posX][posY] = matRes[posX][posY];
}

```

IMPRESIÓN DE PANTALLA DE RESULTADO

[]	[]	[]	[]	[]	[]	[]	[]
[]	[]	[]	[]	[]	[]	[]	[]
[]	[]	[]	[]	x	[]	[]	[]
[]	[]	[]	[]	[]	[]	[]	[]
[]	[]	[]	[]	[]	[]	[]	[]
[]	[]	[]	[]	[]	[]	[]	[]
[]	[]	[]	[]	[]	[]	[]	x
[]	[]	[]	[]	[]	[]	[]	[]

JUEGO CONCENTRECE

Puntaje del Jugador 1: 19

Puntaje del Jugador 2: 12

Jugador 1

Digite Jugada Según la Numeración: (0 para salir del Juego): 56

El Ganador es el << Jugador 1 >> con un puntaje de: 20