

```

#include <iostream>
#include <stdio.h>

using namespace std;

typedef char tablero[9][9];

void TableroIngles(tablero tab){
    int i, j;

    for (i = 0; i < 7; i++)
    {
        for (j = 0; j < 7; j++)
        {
            if (i == 3 && j == 3)
            {
                tab[i][j] = ' ';
            }else{
                if( (i < 2 || i > 4)&&(j < 2 || j > 4) ){
                    tab[i][j] = '.';
                }else{
                    tab[i][j] = 'O';
                }
            }
        }
    }
}

void TableroAleman(tablero tab){
    int i, j;

    for (i = 0; i < 9; i++)
    {
        for (j = 0; j < 9; j++)
        {
            if (i == 4 && j == 4)
            {
                tab[i][j] = ' ';
            }else{
                if( (i < 3 || i > 5)&&(j < 3 || j > 5) ){
                    tab[i][j] = '.';
                }else{
                    tab[i][j] = 'O';
                }
            }
        }
    }
}

```

```

    }
}

void TableroAsimetrico(tablero tab){
    int i, j;

    for (i = 0; i < 8; i++)
    {
        for (j = 0; j < 8; j++)
        {
            if (i == 4 && j == 3)
            {
                tab[i][j] = ' ';
            }else{
                if( (i < 3 || i > 5)&&(j < 2 || j > 4) ){
                    tab[i][j] = '.';
                }else{
                    tab[i][j] = 'O';
                }
            }
        }
    }
}

```

```

void mostrarTableroIngles(tablero tab){
    int fila, columna, aux , y = 1;

    for (fila = 0; fila < 7; fila++)
    {
        if (fila == 0)
        {
            for (aux = 0; aux <= 7; aux++)
            {
                if (aux == 0)
                {
                    cout << " ";
                }else{
                    cout << " " << aux;
                }
            }
            cout << endl;
        }
        cout << y++ << " ";
    }
}

```

```

        for (columna = 0; columna < 7; columna++)
        {
            printf(" %c ", tab[fila][columna]);
        }
        cout << endl;
    }
}

```

```

void mostrarTableroAleman(tablero tab){
    int fila, columna, aux , y = 1;

    for (fila = 0; fila < 9; fila++)
    {
        if (fila == 0)
        {
            for (aux = 0; aux <= 9; aux++)
            {
                if (aux == 0)
                {
                    cout << " ";
                }else{
                    cout << " " << aux;
                }
            }
            cout << endl;
        }
        cout << y++ << " ";

        for (columna = 0; columna < 9; columna++)
        {
            printf(" %c ", tab[fila][columna]);
        }
        cout << endl;
    }
}

```

```

void mostrarTableroAsimetrico(tablero tab){
    int fila, columna, aux , y = 1;

    for (fila = 0; fila < 8; fila++)
    {
        if (fila == 0)
        {
            for (aux = 0; aux <= 8; aux++)

```

```

        {
            if (aux == 0)
            {
                cout << " ";
            }else{
                cout << " " << aux;
            }
        }
        cout << endl;
    }
    cout << y++ << " ";

    for (columna = 0; columna < 8; columna++)
    {
        printf(" %c ", tab[fila][columna]);
    }
    cout << endl;
}
}

```

```

int validarMovimientoIngles(int MoverFichaFila, int
MoverFichaColumna, int DestinoFichaFila, int DestinoFichaColumna,
tablero tab){
    if (tab[MoverFichaFila-1][MoverFichaColumna-1] == 'O')
    {
        if ((MoverFichaFila < 1) || (MoverFichaColumna > 7) ||
(DestinoFichaFila < 1) || (DestinoFichaColumna > 7) ||
(DestinoFichaFila > 7) || (DestinoFichaColumna < 1))
        {
            return 0;
        }else{
            if (MoverFichaFila == DestinoFichaFila &&
MoverFichaColumna == DestinoFichaColumna)
            {
                return 0;
            }else{
                if (MoverFichaFila == DestinoFichaFila ||
MoverFichaColumna == DestinoFichaColumna)
                {
                    if ((MoverFichaFila - DestinoFichaFila == 2 ||
MoverFichaFila - DestinoFichaFila == -2) || (MoverFichaColumna -
DestinoFichaColumna == 2 || MoverFichaColumna -
DestinoFichaColumna == -2))

```

```

        {
            if (tab[DestinoFichaFila -
1][DestinoFichaColumna - 1] == 'O' || tab[DestinoFichaFila -
1][DestinoFichaColumna - 1] == '.')
            {
                return 0;
            }else{
                return 1;
            }
        }else{
            return 0;
        }
    }else{
        return 0;
    }
}

}
}else{
    return 0;
}
}

int validarMovimientoAleman(int MoverFichaFila, int
MoverFichaColumna, int DestinoFichaFila, int DestinoFichaColumna,
tablero tab){
    // que mueva una ficha, no un espacio vacio o una X
    if (tab[MoverFichaFila-1][MoverFichaColumna-1] == 'O')
    {
        //que sea un movimiento dentro del tablero
        if ((MoverFichaFila < 1 && MoverFichaFila > 9) &&
(DestinoFichaFila < 1 && DestinoFichaColumna > 9))
        {
            return 0;
        }else{
            // que no se mueva al mismo lugar donde esta
            if (MoverFichaFila == DestinoFichaFila &&
MoverFichaColumna == DestinoFichaColumna)
            {
                return 0;
            }else{
                // que no se mueva en diagonal
                if (MoverFichaFila == DestinoFichaFila ||
MoverFichaColumna == DestinoFichaColumna)
                {
                    // que se mueva solo 2 espacios
                    if ((MoverFichaFila - DestinoFichaFila == 2 ||
MoverFichaFila - DestinoFichaFila == -2) || (MoverFichaColumna -

```

```
DestinoFichaColumna == 2 || MoverFichaColumna -  
DestinoFichaColumna == -2))  
  
    {  
  
        //que no exista ficha en el lugar de  
destino, ni que sea una esquina prohibida  
        if (tab[DestinoFichaFila -  
1][DestinoFichaColumna - 1] == 'O' || tab[DestinoFichaFila -  
1][DestinoFichaColumna - 1] == '.')  
        {  
  
            return 0;  
  
        }else{  
  
            return 1;  
  
        }  
    }else{  
        return 0;  
    }  
}  
}else{  
    return 0;  
}  
}  
}  
  
}else{  
    return 0;  
}  
}  
}
```

```
int validarMovimientoAsimetrico(int MoverFichaFila, int  
MoverFichaColumna, int DestinoFichaFila, int DestinoFichaColumna,  
tablero tab){  
    // que mueva una ficha, no un espacio vacio o una X  
    if (tab[MoverFichaFila-1][MoverFichaColumna-1] == 'O')  
    {  
  
        //que sea un movimiento dentro del tablero  
        if ((MoverFichaFila < 1) || (MoverFichaColumna > 8) ||  
(DestinoFichaFila < 1) || (DestinoFichaColumna > 8) ||  
(DestinoFichaFila > 8) || (DestinoFichaColumna < 1))  
        {  
  
            return 0;  
        }else{  
            // que no se mueva al mismo lugar donde esta  
            if (MoverFichaFila == DestinoFichaFila &&  
MoverFichaColumna == DestinoFichaColumna)  
            {  
  
                return 0;  
            }else{  
  
                // que no se mueva en diagonal
```



```

        if (MoverFichaColumna == DestinoFichaColumna)
        {
            if (MoverFichaFila < DestinoFichaFila)
            {
                tab[MoverFichaFila][MoverFichaColumna - 1] = ' ';
            }else{
                tab[MoverFichaFila - 2][MoverFichaColumna - 1] = '
';
            }
        }
    }
}

```

```

int Ganar(tablero tab){
    int fila, columna, cuentaFichas = 0;

    for (fila= 0; fila < 9; fila++)
    {
        for (columna = 0; columna < 9; columna++)
        {
            if(tab[fila][columna] == 'O')
            {
                cuentaFichas++;
            }
        }
    }

    if (cuentaFichas == 1)
    {
        return 1;
    }else{
        return 0;
    }
}

```

```

void JugarIngles(){
    int MoverFichaFila, MoverFichaColumna, DestinoFichaFila,
DestinoFichaColumna, movValido, salida = 0, ganado;
    tablero tab;
    TableroIngles(tab);
    mostrarTableroIngles(tab);
    ganado = Ganar(tab);
}

```



```

    while (salida != 1){
        cout << "Ingrese la posicion de origen (fila, columna):" <<
endl;
        cout << "Numero de fila: ";
        // pasa a todas las funciones
        scanf(" %i", &MoverFichaFila);
        cout << "Numero de columna: ";
        scanf(" %i", &MoverFichaColumna);
        cout << endl;

        cout << "Ingrese la posicion de destino(fila, columna):" <<
endl;
        cout << "Numero de fila: ";
        scanf(" %i", &DestinoFichaFila);
        cout << "Numero de columna: ";
        scanf(" %i", &DestinoFichaColumna);
        cout << endl;

        movValido = validarMovimientoIngles(MoverFichaFila,
MoverFichaColumna, DestinoFichaFila, DestinoFichaColumna, tab);
        if (movValido == 1)
        {
            mover(MoverFichaFila, MoverFichaColumna,
DestinoFichaFila, DestinoFichaColumna, tab);
            mostrarTableroIngles(tab);
        }else{
            cout << "Movimiento no valido" << endl;
        }

        ganado = Ganar(tab);
        if (salida != 1)
        {
            if (ganado == 1)
            {
                salida = 1;
                cout << "Fin del juego";
            }
        }else{
        }
    }
}

```

```

void JugarAleman(){
    int MoverFichaFila, MoverFichaColumna, DestinoFichaFila,
DestinoFichaColumna, movValido, salida = 0, ganado;
    tablero tab;
    TableroAleman(tab);
    mostrarTableroAleman(tab);
    ganado = Ganar(tab);

    while (salida != 1){
        cout << "Ingrese la posicion de origen (fila, columna):" <<
endl;
        cout << "Numero de fila: ";
        // pasa a todas las funciones
        scanf(" %i", &MoverFichaFila);
        cout << "Numero de columna: ";
        scanf(" %i", &MoverFichaColumna);
        cout << endl;

        cout << "Ingrese la posicion de destino(fila, columna):" <<
endl;
        cout << "Numero de fila: ";
        scanf(" %i", &DestinoFichaFila);
        cout << "Numero de columna: ";
        scanf(" %i", &DestinoFichaColumna);
        cout << endl;

        movValido = validarMovimientoAleman(MoverFichaFila,
MoverFichaColumna, DestinoFichaFila, DestinoFichaColumna, tab);
        if (movValido == 1)
        {
            mover(MoverFichaFila, MoverFichaColumna,
DestinoFichaFila, DestinoFichaColumna, tab);
            mostrarTableroAleman(tab);
        }else{
            cout << "Movimiento no valido" << endl;
        }

        ganado = Ganar(tab);
        if (salida != 1)
        {
            if (ganado == 1)

```

```

        {
            salida = 1;
            cout << "Fin del juego";
        }
    }else{

    }

}

}

}

void JugarAsimetrico(){
    int MoverFichaFila, MoverFichaColumna, DestinoFichaFila,
DestinoFichaColumna, movValido, salida = 0, ganado;
    tablero tab;
    TableroAsimetrico(tab);
    mostrarTableroAsimetrico(tab);
    ganado = Ganar(tab);

    while (salida != 1){
        cout << "Ingrese la posicion de origen (fila, columna):" <<
endl;
        cout << "Numero de fila: ";
        // pasa a todas las funciones
        scanf(" %i", &MoverFichaFila);
        cout << "Numero de columna: ";
        scanf(" %i", &MoverFichaColumna);
        cout << endl;

        cout << "Ingrese la posicion de destino(fila, columna):" <<
endl;
        cout << "Numero de fila: ";
        scanf(" %i", &DestinoFichaFila);
        cout << "Numero de columna: ";
        scanf(" %i", &DestinoFichaColumna);
        cout << endl;

        movValido = validarMovimientoAsimetrico(MoverFichaFila,
MoverFichaColumna, DestinoFichaFila, DestinoFichaColumna, tab);
        if (movValido == 1)
        {
            mover(MoverFichaFila, MoverFichaColumna,
DestinoFichaFila, DestinoFichaColumna, tab);

```

```

        mostrarTableroIngles(tab);
    }else{
        cout << "Movimiento no valido" << endl;
    }

    ganado = Ganar(tab);
    if (salida != 1)
    {
        if (ganado == 1)
        {
            salida = 1;
            cout << "Fin del juego";
        }
    }else{

    }
}

}

int main() {
    int modelo;

    cout << "Menu - Juego Senku" << endl;
    cout << "-----" << endl;
    cout << "1. Estilo Ingles" << endl;
    cout << "2. Estilo Aleman" << endl;
    cout << "3. Estilo Asimetrico" << endl;
    cout << "-----" << endl;
    cout << "0. Salir del programa" << endl << endl;

    cout << "Seleccionar la opcion: ";

    while (modelo > 3 || modelo < 0){
        cin >> modelo;
        if (modelo > 3 || modelo < 0){
            cout << "Ingrese un numero valido de la lista: " <<
endl;
        }
    }
    switch (modelo)
    {
        case 1:
            cout << endl << "Tablero Ingles: " << endl << endl;

```

```
        JugarIngles();  
        break;  
  
    case 2:  
        cout << endl << "Tablero Aleman: " << endl << endl;  
        JugarAleman();  
        break;  
  
    case 3:  
        cout << endl << "Tablero Asimetrico: " << endl << endl;  
        JugarAsimetrico();  
  
    case 0:  
        cout << "Adios";  
        break;  
    }  
    return 0;  
}
```