```
#include <iostream>
#include<string>
using namespace std;
//Matriz triangular
int ratificar_salida_ejecutar(int senbih[][22], int datoy2,int datox2, int datoy1, int
datox1);//Me permite ver mis restricciones sorbre mi destino y si cumple ejecuta
int pedir_entrada(int senbih[][22],int &datoy1, int &datox1);//pide mi coordenada
int ratificar_entrada(int senbih[][22],int  datoy1, int datox1);//verifica que mi
coordenada de origen este bien
void convertidor(int senbih[][22],string senkuh[][18],int nfilah, int ncolumh);//convierte
mis valores de mi ta
int contar_senbih(int senbih[][22],int nfilah, int ncolumh);//me permite contar la
cantidad de fichas en el tablero para asi determinar si se gana
int contad_jugadas(int senbih[][22],int nfilah, int ncolumh);//sirve para contar las
jugadas y saber si para el programa
int pedir_salida(int senbih[][22], int &datoy2, int &datox2, int datoy1,int datox1);//me
sirve para pedir mi destino y pasa a traves de un filtro minimo y luego de uno mas fuerte
void imprimirh(string senkuh[][18]);//imprime mis valores de matriz

//Matriz suma
void classicsenk();
int contador_jugadas(int senbi[][17],int nfil, int ncolum);// funcion que me cuenta
jugadas(restriccion)
void imprimirsnk(string senku[][14], int nfil,int ncolum);//funcion que imprime la matriz
senku
void cambiador(int senbi[][17], string senku[][14],int nfil ,int ncolum);//redefine las
variables segun la
int preguntar_origen(int senbi[][17],int &coordy1,int &coordx1);//pregunta por mi
coordenada de origen
int preguntar_destino(int senbi[][17], int &coordy2,int &coordx2, int coordy1, int
coordx1);//pregunta por mi coordenada de desitno
int comprobar_origen(int senbi[][17],int coordy1, int coordx1);//comprueba que cumpla
todas las restricciones
int comprobar_destino_jugada(int senbi[][17], int coordy2, int coordx2, int coordy1, int
coordx1);//comprueba las ultimas restricciones y ejecuta la jugada
int piezas(int senbi[][17], int nfil, int ncolum);//cuenta las fichas con valor 1 en el
senbi para asi poder saber las fichas en tablero que se pueden mover

int main() {
  cout << "Menú Juego Senku" <<endl;
  cout << "----------------------------"<< endl;
  cout << "Elija modo de juego"<<endl;
  cout << "1.- Estilo triangular" << endl;
  cout << "2.- Estilo ingles"<< endl;
  cout << "----------------------------"<< endl;
  int c;
  cin >> c;
    if (c == 1){
      const int ncolumh = 22;
      const int nfilah = 10;
      int datoy1;
      int datox1;
      int datoy2;
      int datox2;
```

```
      bool t = true;
      bool end = true;

      string senkuh[10][18]={{"  ","1","   ","2 ","  ","3 ","  ","4 ","  ","5 ","  ","6
","  ","7 ","  ","8 ","  ","9 "},
                             {"1 "," ","  ","  ","  ","  ","  ","  ","  "," ○ "," "," ","
","  ","  ","  ","  ","  "},
                             {"   "," ","  ","  ","  ","  ","  ","  ","/","   ","\"," "," ","
","  ","  ","  ","  "},
                             {"2 "," ","  ","  ","  ","  ","  "," ● ","-","---","-"," ●
","  ","  ","  ","  ","  ","  "},
                             {"   "," ","  ","  ","  ","  ","/","   ","\","   ","/","
","\","  ","  ","  ","  ","  "},
                             {"3 "," ","  ","  ","  "," ● ","-","---","-"," ● ","-","---
","-"," ● ","  ","  ","  ","  "},
                             {"   "," ","  ","  ","/","   ","\","   ","/","   ","\","
","/","   ","\","  ","  ","  "},
                             {"4  "," ","  ","  "," ● ","-","---","-"," ● ","-","---","-"," ●
","-","---","-"," ●","  ","  "},
                             {"   "," ","/","   ","\","   ","/","   ","\","   ","/","
","\","   ","/","   ","\","  "},
                             {"5"," ● ","-","---","-"," ● ","-","---","-"," ● ","-","---
","-"," ● ","-","---","-"," ● "}};

      int senbih[13][22]={{2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
                          {2,2,2,2,2,2,2,2,2,0,2,2,2,2,2,2,2,2,2,2,2,2},
                          {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
                          {2,2,2,2,2,2,2,1,2,2,2,1,2,2,2,2,2,2,2,2,2,2},
                          {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
                          {2,2,2,2,2,1,2,2,2,1,2,2,2,1,2,2,2,2,2,2,2,2},
                          {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
                          {2,2,2,1,2,2,2,1,2,2,2,1,2,2,2,1,2,2,2,2,2,2},
                          {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
                          {2,1,2,2,2,1,2,2,2,1,2,2,2,1,2,2,2,1,2,2,2,2},
                          {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},

{2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},{2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2
},
                          };

      while(t == true){
    convertidor(senbih,senkuh,nfilah,ncolumh);
    imprimirh(senkuh);
        if(contar_senbih(senbih,nfilah,ncolumh) == 1){
            break;
    }
        if(contad_jugadas(senbih,nfilah,ncolumh) <= 0){
        end = false;
          break;
    }
      pedir_entrada(senbih,datoy1,datox1);
      pedir_salida(senbih,datoy2,datox2,datoy1,datox1);
  }
        if(end == true){
```

```
    cout << "You win congrats" << endl;
  }
        else{
    cout << "Youu looose" << endl;
  }
  cout <<"Thx 4 playing";
  }
    else if (c==2){
      const int nfil = 17;
      const int ncolum =17;
      bool q = true;
      bool rng = false;
      int coordy1 = 2;//representa mi coordenada de origen en el eje y(fila)
      int coordx1 = 4;//representa mi coordenada de destino en el eje x(columna)
      int coordy2 = 0;//representa mi coordenada de origen en el eje y(fila)
      int coordx2 = 0;//representa mi coordenada de destino en el eje x(columna)

      string senku[14][14]={{"   "," 1 ","   "," 2 ","   "," 3 ","   "," 4 ","   "," 5 ","
","," 6 ","   "," 7 "},
                            {" 1 ","   ","   ","   ","   "," ● ","---"," ● ","---"," ●
","   ","   ","   ","   "},
                            {"   ","   ","   ","   ","   "," | ","   "," | ","   "," | ","
","   ","   ","   "},
                            {" 2 ","   ","   ","   ","   "," ● ","---"," ● ","---"," ●
","   ","   ","   ","   "},
                            {"   ","   ","   ","   ","   "," | ","   "," | ","   "," | ","
","   ","   ","   "},
                            {" 3 "," ● ","---"," ● ","---"," ● ","---"," ● ","---","
● ","---"," ● ","---"," ● "},
                            {"   "," | ","   "," | ","   "," | ","   "," | ","   "," | ","
","   "," | "},
                            {" 4 "," ● ","---"," ● ","---"," ● ","---"," ○ ","---","
● ","---"," ● ","---"," ● "},
                            {"   "," | ","   "," | ","   "," | ","   "," | ","   "," | ","
","   "," | "},
                            {" 5 "," ● ","---"," ● ","---"," ● ","---"," ● ","---","
● ","---"," ● ","---"," ● "},
                            {"   ","   ","   ","   ","   "," | ","   "," | ","   "," | ","
","   ","   ","   "},
                            {" 6 ","   ","   ","   ","   "," ● ","---"," ● ","---"," ●
","   ","   ","   ","   "},
                            {"   ","   ","   ","   ","   "," | ","   "," | ","   "," | ","
","   ","   ","   "},
                            {" 7 ","   ","   ","   ","   "," ● ","---"," ● ","---"," ●
","   ","   ","   ","   "}};
      int senbi[nfil][ncolum]={ {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
                                {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
                                {2,2,2,2,2,2,1,2,1,2,1,2,2,2,2,2,2},
                                {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
                                {2,2,2,2,2,2,1,2,1,2,1,2,2,2,2,2,2},
                                {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
                                {2,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2,2},
                                {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
                                {2,2,1,2,1,2,1,2,0,2,1,2,1,2,1,2,2},
```

```
                                {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
                                {2,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2},
                                {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
                                {2,2,2,2,2,2,1,2,1,2,1,2,2,2,2,2},
                                {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
                                {2,2,2,2,2,2,1,2,1,2,1,2,2,2,2,2},
                                {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
                                {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2} };


      while(q == true ){

        cambiador(senbi,senku,nfil,ncolum);
        imprimirsnk(senku,nfil,ncolum);
        if(piezas(senbi,nfil,ncolum) == 1){//en caso cuente una ficha significa que quedo
una pieza y por lo tanto ganaste
          rng = true;
          break;
        }
        if(contador_jugadas(senbi,nfil,ncolum) == 0 ){//si ya no hay jugadas posibles el
juego finzaliza y pierdes
          rng = false;
          break;
        }
        preguntar_origen(senbi, coordy1, coordx1);
        preguntar_destino(senbi, coordy2, coordx2,coordy1,coordx1);
      }
      if( rng == true){
        cout << "You win" <<endl;
      }
      else{
        cout <<" You lost" <<endl;
      }
      cout << "Thx 4 playing"<<endl;



      cout << endl;

  }
  }

  // funciones triangular
  void convertidor(int senbih[][22],string senkuh[][18],int nfilah, int ncolumh){
    for(int i = 0; i < nfilah; i++){
      for(int j = 0; j < ncolumh; j++){
        if(senbih[i][j]==1){
          senkuh[i][j] =" 🔴 ";

        }
        if(senbih[i][j] == 0){
          senkuh[i][j] =" ◯ ";
        }
      }
```

```
    }

  }
  int contar_senbih(int senbih[][22],int nfilah, int ncolumh){
    int contador_h = 0;
    for(int i = 0; i < nfilah ; i++){
      for( int j = 0; j< ncolumh; j++){
        if(senbih[i][j] == 1){
          contador_h++;
        }
      }
    }
    return contador_h;
  }

  int pedir_entrada(int senbih[][22],int &datoy1, int &datox1){//con ampersand para asi
mantener los datos
    bool p = true;
    cout <<"Ingrese coordenada de origen" << endl;
    while(p == true){
      cin >> datoy1;
      cin >> datox1;
      if(senbih[2*datoy1 - 1][2*datox1 - 1] == 1){// sim cumple entra ya que este es el
minimo requisito para evaluarlo
        if(ratificar_entrada(senbih,datoy1,datox1) != 0){
          p = false;
        }
        else{
          cout << "Verificar valor" << endl;
        }
      }
      if(senbih[2*datoy1 - 1][2*datox1 - 1] == 0){
        cout << "Coordenada con valor vacio reingrese"<<endl;//bota esto porque no cumple
con la funcion principal de ser 1 dentro de la matriz senbih
      }
      if(senbih[2*datoy1 - 1][2*datox1 - 1] == 2){
        cout << "Coordenada no localizada"<<endl;
      }
    }
    return 0;
  }
  int ratificar_entrada(int senbih[][22],int  datoy1, int datox1){
    int account = 0;

    if(senbih[(2*datoy1 - 1)][(2*datox1 - 1) + 4] == 1){
      if(senbih[(2*datoy1 - 1)][(2*datox1 - 1) + 8] == 0){
        account++;
      }
    }
    if(senbih[(2*datoy1 - 1)][(2*datox1 - 1) - 4] == 1){
      if(senbih[(2*datoy1 - 1)][(2*datox1 - 1) - 8] == 0){
        account++;
      }
    }
```

```
    if(senbih[(2*datoy1 - 1) - 2][(2*datox1 - 1) + 2] == 1){
      if(senbih[(2*datoy1 - 1) - 4][(2*datox1 - 1) + 4] == 0){
        account++;
      }
    }
    if(senbih[(2*datoy1 - 1)-2][(2*datox1 - 1) - 2] == 1){
      if(senbih[(2*datoy1 - 1) - 4][(2*datox1 - 1) - 4] == 0){
        account++;
      }
    }
    if(senbih[(2*datoy1 - 1) + 2][(2*datox1 - 1) + 2] == 1){
      if(senbih[(2*datoy1 - 1) + 4][(2*datox1 - 1) + 4] == 0){
        account++;
      }
    }
    if(senbih[(2*datoy1 - 1) + 2][(2*datox1 - 1) - 2] == 1){
      if(senbih[(2*datoy1 - 1)  + 4][(2*datox1 - 1) - 4] == 0){
        account++;
      }
    }
    return account;
  }
  int pedir_salida(int senbih[][22], int &datoy2, int &datox2, int datoy1,int datox1){
    bool q = true;
    cout << "Ingrese su coordenada de destino" << endl;
    while( q == true){
      cin >> datoy2;
      cin >> datox2;
      if(senbih[2*datoy2 - 1][2*datox2 - 1] == 0){
        if(ratificar_salida_ejecutar(senbih,datoy2,datox2,datoy1,datox1) == 1){
          q = false;
        }
        else{
          cout << "Verificar coordenada" << endl;
        }
      }
      if(senbih[2*datoy2 - 1][2*datox2 - 1] == 2){
        cout << "Coordenada no localizada"<<endl;
      }
    }
    return 0;
  }
  int ratificar_salida_ejecutar(int senbih[][22], int datoy2,int datox2, int datoy1, int
datox1){
    int s = 0;
    if(datoy2 == datoy1){
      if(abs(datox2-datox1) == 4){
        senbih[2*datoy2 - 1][2*datox2 - 1] = 1;
        senbih[2*datoy1 - 1][2*datox1 - 1] = 0;
        senbih[2*datoy2 - 1][(datox1+datox2) - 1] =  0;
        s = 1;

      }
    }
```

```
  if((datoy1-datoy2 == 2) && (datox1-datox2 == -2)){//sube diagonal derecha
      senbih[2*datoy2 - 1][2*datox2 - 1] = 1;
      senbih[2*datoy1 - 1][2*datox1 - 1] = 0;
      senbih[datoy2+datoy1 - 1][(datox1+datox2) - 1] =  0;
      s = 1;
  }
  if((datoy1-datoy2 == 2) && (datox1-datox2 == 2)){//sube diagonal izquierda
      senbih[2*datoy2 - 1][2*datox2 - 1] = 1;
      senbih[2*datoy1 - 1][2*datox1 - 1] = 0;
      senbih[datoy2+datoy1 - 1][(datox1+datox2) - 1] =  0;
      s = 1;
  }
  if((datoy1-datoy2 == -2) && (datox1-datox2 == 2)){//baja diagonal izquierda
      senbih[2*datoy2 - 1][2*datox2 - 1] = 1;
      senbih[2*datoy1 - 1][2*datox1 - 1] = 0;
      senbih[datoy2+datoy1 - 1][(datox1+datox2) - 1] =  0;
      s = 1;
  }
  if((datoy1-datoy2 == -2) && (datox1-datox2 == -2)){//baja diagonal derecha
      senbih[2*datoy2 - 1][2*datox2 - 1] = 1;
      senbih[2*datoy1 - 1][2*datox1 - 1] = 0;
      senbih[datoy2+datoy1 - 1][(datox1+datox2) - 1] =  0;
      s = 1;
  }
  return s;
}
int contad_jugadas(int senbih[][22],int nfilah, int ncolumh){
  int jugadas= 0;

  for(int i = 0; i < nfilah ;i++ ){
    for(int j = 0; j < ncolumh; j++){
      if(senbih[i][j] == 1){
        if(senbih[i][j+4] == 1){
          if(senbih[i][j+8] == 0){
            jugadas++;
          }
        }

        if(senbih[i][j-4] == 1){
          if(senbih[i][j-8] == 0){
            jugadas++;
          }
        }

        if(senbih[i+2][j+2] == 1){
          if(senbih[i+4][j+4] == 0){
            jugadas++;
          }
        }

        if(senbih[i+2][j-2] == 1){
          if(senbih[i+4][j-4] == 0){
            jugadas++;
          }
```

```
          }

          if(senbih[i-2][j+2] == 1){
            if(senbih[i-4][j+4] == 0){
              jugadas++;
            }
          }

          if(senbih[i-2][j-2] == 1){
            if(senbih[i-4][j-4] == 0){
              jugadas++;
            }
          }
        }
      }
    }

    return jugadas;
  }
  void imprimirh(string senkuh[][18]){
    for(int i = 0; i < 10 ; i++){
      for( int j = 0; j < 18; j++){
        cout << senkuh[i][j];
      }
      cout << endl;
    }
  }
  // funciones cruz
  int contador_jugadas(int senbi[][17],int nfil , int ncolum){// funcion que me cuenta
jugadas(restriccion)
    int cont1 = 0;
    int cont2 = 0;
    int cont3 = 0;//contadores para determinar jugadas
    int cont4 = 0;

    for(int i = 0;i<nfil;i++){
      for(int j = 0;j<ncolum;j++){
        if(senbi[i][j] == 1){
          if(senbi[i][j+2]==1){
            if(senbi[i][j+4]==0){
              cont1++;
            }
          }
        }
        if(senbi[i][j] == 1){
          if(senbi[i][j-2]==1){
            if(senbi[i][j-4]==0){
              cont2++;
            }
          }
        }
        if(senbi[i][j] == 1){
          if(senbi[i+2][j]==1){
            if(senbi[i+4][j]==0){
```

```
            cont3++;
          }
        }
      }
      if(senbi[i][j] == 1){
        if(senbi[i-2][j]==1){
          if(senbi[i-4][j]==0){
            cont4++;
          }
        }
      }
    }
  }
  return cont1 + cont2 + cont3 + cont4;
}
int piezas(int senbi[][17], int nfil, int ncolum){
  int contador_1 = 0;//cuenta las piezas en el tablero
  for(int i = 0; i < nfil; i++){
    for( int j = 0; j < ncolum; j++){
      if(senbi[i][j] == 1){
        contador_1++;
      }
    }
  }
  return contador_1;
}
void cambiador(int senbi[][17], string senku[][14],int nfil ,int ncolum){//redefine las
variables segun la segunda matriz

  for(int i = 0;i< nfil;i++){
    for(int j = 0; j < ncolum ; j++){
      if(senbi[i][j] == 1){
        senku[i-1][j-1] = " ◍ ";
      }
      if(senbi[i][j] == 0){
        senku[i-1][j-1] = " ◯ ";
      }
    }
  }
}
void imprimirsnk(string senku[][14],int nfil,int ncolum){
  for(int i = 0;i< (nfil-3) ;i++){
    for(int j = 0; j < (ncolum-3) ; j++){
      cout << senku[i][j];
    }
    cout << endl;
  }
}
int preguntar_origen(int senbi[][17],int &coordy1, int &coordx1){
  bool p = true;
  cout <<"Ingrese su coordenada(par) de origen"<<endl;
  while( p ==true){
    cin>> coordy1;
    cin>> coordx1;
```

```
      if(senbi[coordy1*2][coordx1*2] == 1){
        if(comprobar_origen(senbi,coordy1,coordx1) != 0){
          p = false;
        }
        else{
          cout << "Verifique su entrada"<<endl;
        }
      }
      if(senbi[2*coordy1][2*coordx1] == 0){
        cout << "Casilla seleccionada vacía reingrese valor"<<endl;
      }
      if(senbi[coordy1*2][coordx1*2] == 2){
        cout << "Coordenada incorrecta reingrese valor" <<endl;
      }
    }
    return 0;

  }
  int preguntar_destino(int senbi[][17],int &coordy2,int &coordx2,int coordy1, int
coordx1){
    bool r = true;
    cout << "Ingrese coordenada(par) de destino" <<endl;
    while(r == true){
      cin >> coordy2;
      cin >> coordx2;
      if(senbi[2*coordy2][2*coordx2] == 0){
          if(comprobar_destino_jugada(senbi,coordy2,coordx2,coordy1,coordx1) == 1){
            r = false;
          }
          else{
            cout << "Verifique su coordenada" << endl;
          }
        }
      if(senbi[2*coordy2][2*coordx2] == 1){
        cout << "Casilla seleccionada llena reingrese valor"<<endl;
      }
      if(senbi[coordy2*2][coordx2*2] == 2){
        cout << "Coordenada incorrecta reingrese valor" <<endl;
      }
    }
    return 0;
  }
  int comprobar_origen(int senbi[][17],int coordy1, int coordx1){
    int t = 0;
    if(senbi[2*coordy1 + 2][2*coordx1] == 1){
      if(senbi[2*coordy1 + 4][2*coordx1] == 0){
        t++;
      }
    }
    if(senbi[2*coordy1 - 2][2*coordx1] == 1){
      if(senbi[2*coordy1 - 4][2*coordx1] == 0){
        t++;
      }
    }
```

```
    if(senbi[2*coordy1][2*coordx1 + 2] == 1){
      if(senbi[2*coordy1][2*coordx1 + 4] == 0){
        t++;
      }
    }
    if(senbi[2*coordy1][2*coordx1 - 2] == 1){
      if(senbi[2*coordy1][2*coordx1 - 4] == 0){
        t++;
      }
    }
    return t;
  }
  int comprobar_destino_jugada(int senbi[][17], int coordy2, int coordx2, int coordy1, int
coordx1){
    int w = 0;// valores qeu retornar para alguna condicional
    if(((coordy2 == coordy1) && (coordx2 != coordx1)) || ((coordy2 != coordy1) && (coordx2
== coordx1))){
      if(coordy2 == coordy1){
        if(abs(coordx2 - coordx1) == 2){
          senbi[2*coordy2][2*coordx2] = 1;
          senbi[2*coordy1][2*coordx1] = 0;
          senbi[2*coordy2][(2*coordx1 + 2*coordx2)/2] = 0;//en caso ingrese se ejecuta la
jugada la cual es luego leida por otra funcion
          w = 1;
        }
      }
      else{
        if(abs(coordy2 - coordy1) == 2){
          senbi[2*coordy1][2*coordx1] = 0;
          senbi[2*coordy2][2*coordx2] = 1;
          senbi[coordy1 + coordy2][2*coordx2] = 0;
          w = 1;
        }

      }
    }
    return w;
  }
```