

```

#include <iostream>
#include <fstream>

using namespace std;

struct Tavola {
    char *v;
    int dim;
};

struct elem {
    int r;
    int c;
    elem *pun;
};

void inizializzaTavola(Tavola &T, int n) {
    if (n <= 0)
        return;

    T.v = new char[n * n];
    T.dim = n;

    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            T.v[i * n + j] = '-';
}

bool contrassegna(Tavola &T, int i, int j, char s) {
    if (i >= T.dim || j >= T.dim || i<0 || j<0)
        return false;

    if (s != 'X' && s != 'O')
        return false;

    if (T.v == NULL)
        return false;

    if (T.v[i * T.dim + j] != '-')
        return false;

    T.v[i * T.dim + j] = s;
    return true;
}

bool filettoRiga(Tavola &T, char s) {
    for (int i = 0; i < T.dim; i++) {
        for (int j = 0; j < T.dim - 2; j++) {
            if (T.v[i * T.dim + j] == s && T.v[i * T.dim + j + 1]
                return true;
            }
        }

    return false;
}

```

```

bool filettoColonna(Tavola &T, char s) {

    for (int j = 0; j < T.dim; j++) {
        for (int i = 0; i < T.dim - 2; i++) {
            if (T.v[i * T.dim + j] == s && T.v[(i + 1) * T.dim +
                return true;
            }
        }

    return false;
}

void inserisci_testa(elem *&testa, int r, int c) {
    elem *p = new elem;
    p->r = r;
    p->c = c;
    p->pun = testa;
    testa = p;
}

// Funzione ausiliaria, non c'era nel testo del compito
void stampa_lista(elem *testa){
    elem *q = testa;
    while(q!=NULL){
        cout << '[' << q->r << ", " << q->c << ']'<< endl;
        if(q->pun != NULL)
            cout << " -> ";
        q = q->pun;
    }
    cout << endl;
}

elem *creaListaLibere(Tavola &T) {
    elem *testa = NULL;

    for (int i = 0; i < T.dim; i++)
        for (int j = 0; j < T.dim; j++)
            if (T.v[i * T.dim + j] == '-')
                inserisci_testa(testa, i, j);

    return testa;
}

elem *creaListaOccupate(Tavola &T) {
    elem *testa = NULL;

    for (int i = 0; i < T.dim; i++)
        for (int j = 0; j < T.dim; j++)
            if (T.v[i * T.dim + j] != '-')
                inserisci_testa(testa, i, j);

    return testa;
}

void stampaTavola(Tavola &T) {
    cout << "Tavola" << endl;
}

```

```

    if(T.v == NULL) {
        cout << "vuota!!" << endl;
        return;
    }
    for (int i = 0; i < T.dim; i++) {
        for (int j = 0; j < T.dim; j++)
            cout << T.v[i * T.dim + j] << " ";
        cout << endl;
    }
}

void stampaTavola(Tavola &T, const char *str) {
    if(T.v == NULL)
        return;

    ofstream out(str);
    if (!out)
        return;

    for (int i = 0; i < T.dim; i++) {
        for (int j = 0; j < T.dim; j++)
            out << T.v[i * T.dim + j] << " ";
        out << endl;
    }
}

void cancella(Tavola &T){
    delete[] T.v;
    T.v = NULL;
    T.dim = 0;
}

int main() {
    int dim;
    bool b;
    char c;

    // immettere la dimensione
    cout << "Inserisci la dimensione: ";
    cin >> dim; // inseriamo 5

    //creazione di una tavola 5x5
    Tavola T;
    inizializzaTavola(T, dim);
    //stampa della tavola
    stampaTavola(T);

    contrassegna(T, 0, 0, 'X');
    contrassegna(T, 1, 1, 'X');
    contrassegna(T, 2, 1, 'X');
    contrassegna(T, 3, 1, 'X');
    contrassegna(T, 6, 0, 'X');
    contrassegna(T, 2, 2, 'O');
    contrassegna(T, 2, 4, 'O');
    contrassegna(T, 4, 3, 'X');

```

```

//stampa della tavola
stampaTavola(T);

//crea lista con le posizioni delle caselle occupate e libere
elem *lo = creaListaOccupate(T);
elem *ll = creaListaLibere(T);

cout << "Lista occupate: ";
stampa_lista(lo);
cout << "Lista libere: ";
stampa_lista(ll);

// controlla se filetto di X in una colonna
c = 'X';
b = filettoColonna(T, c);
if (b)
    cout << c << ": Filetto Colonna" << endl;
else
    cout << c << ": Non Filetto Colonna" << endl;

c = 'O';
b = filettoRiga(T, c);
if (b)
    cout << c << ": Filetto Riga" << endl;
else
    cout << c << ": Non Filetto Riga" << endl;

cancella(T);
stampaTavola(T);

return 0;
}

```