

```

#include <iostream>

using namespace std;

struct Murales{
    char *mat;
    int dim;
};

struct elem{
    int i;
    int j;
    elem *pun;
};

void inizializzaMurales(Murales &M, int N){

    if(N<4)
        M.dim = 4;
    else
        M.dim = N;

    // Creo matrice in memoria dinamica
    M.mat = new char[M.dim * M.dim];

    // Inizializzo matrice con tutte le caselle non colorate
    for(int i=0;i<M.dim;i++)
        for(int j=0;j<M.dim;j++)
            M.mat[i*M.dim+j] = '-';
}

bool coloraCasella(Murales &M, char c, int i, int j){

```

```

//controllo che la matrice sia inizializzata
if(M.mat == NULL)
    return false;

//controllo se il colore passato è valido
if(c!='R' && c!='B' && c!='G')
    return false;

//controllo se non sto uscendo fuori dalla matrice
if(i<0 || j<0 || i>=M.dim || j>=M.dim)
    return false;

int n = M.dim;

//controllo angolo in alto a sinistra
if(i==0 && j==0)
    if(M.mat[i*n+j+1] == c || M.mat[(i+1)*n+j] == c)
        return false;

//controllo angolo in basso a sinistra
if(i==n-1 && j==0)
    if(M.mat[i*n+j+1] == c || M.mat[(i-1)*n+j] == c)
        return false;

//controllo angolo in alto a destra
if(i==0 && j==n-1)
    if(M.mat[i*n+j-1] == c || M.mat[(i+1)*n+j] == c)
        return false;

//controllo angolo in basso a destra
if(i==n-1 && j==n-1)
    if(M.mat[i*n+j-1] == c || M.mat[(i-1)*n+j] == c)
        return false;

```

```

//controllo lato superiore
if(i==0)
    if(M.mat[i*n+j-1] == c || M.mat[i*n+j+1] == c || M.mat[(i+1)*n+j] == c)
        return false;

//controllo lato inferiore
if(i==n-1)
    if(M.mat[i*n+j-1] == c || M.mat[i*n+j+1] == c || M.mat[(i-1)*n+j] == c)
        return false;

//controllo lato sinistro
if(j==0)
    if(M.mat[(i+1)*n+j] == c || M.mat[(i-1)*n+j] == c || M.mat[i*n+j+1] == c)
        return false;

//controllo lato destro
if(j==n-1)
    if(M.mat[(i+1)*n+j] == c || M.mat[(i-1)*n+j] == c || M.mat[i*n+j-1] == c)
        return false;

//controllo casella qualunque
if(M.mat[(i+1)*n+j] == c || M.mat[(i-1)*n+j] == c || M.mat[i*n+j+1] == c || M.mat[i*n+j-1] == c)
    return false;

//finalmente posso cambiare colore
M.mat[i*n+j] = c;
return true;
}

bool coloraPrimaCasella(Murales &M, char c){

    if(M.mat == NULL)
        return false;

```

```

    if(c!='R' && c!='B' && c!='G')
        return false;

    for(int i=0;i<M.dim;i++){
        for(int j=0;j<M.dim;j++){
            if(M.mat[i*M.dim+j]=='-' && coloraCasella(M, c, i, j))
                return true;
        }
    }

    return false;
}

int riempiMurales(Murales &M, char c){

    if(M.mat == NULL)
        return false;

    if(c!='R' && c!='B' && c!='G')
        return false;

    int conta = 0;

    for(int i=0;i<M.dim;i++){
        for(int j=0;j<M.dim;j++){
            if(M.mat[i*M.dim+j]=='-' && coloraCasella(M, c, i, j))
                conta++;
        }
    }

    return conta;
}

```

```

void stampaMurales(Murales &M){

    if(M.mat == NULL)
        return;

    for(int i=0;i<M.dim;i++){
        for(int j=0;j<M.dim;j++){
            cout << M.mat[i*M.dim+j] << ' ';
        }
        cout << endl;
    }
    cout << endl;
}

elem* listaPosizioni(Murales &M, char c){
    elem *testa = NULL;

    if(M.mat == NULL)
        return testa;

    if(c!='R' && c!='B' && c!='G')
        return testa;

    for(int i=0;i<M.dim;i++){
        for(int j=0;j<M.dim;j++){
            if(M.mat[i*M.dim+j] == c) {
                elem *q = new elem;
                q->i = i;
                q->j = j;
                q->pun = testa;
                testa = q;
            }
        }
    }
}

```

```

    return testa;
}

void stampaLista(elem* testa){

    for(elem *q = testa; q!=NULL; q=q->pun){
        cout << '<' << q->i << ',' << q->j << '>';
        if(q->pun!=NULL)
            cout << ", ";
    }
    cout << endl;
}

char* f2(char* str1, char* str2){

    char *str = new char[strlen(str1)+strlen(str2)+1];

    strcpy(str, str1);
    strcpy(&str[strlen(str1)], str2);

    return str;
}

void f3(int N){
    if(N<=1){
        cout << "*" << endl;
        return;
    }

    f3(N-1);

    for(int i=0; i<N-1; i++){
        cout << " ";
    }
}

```

```

    cout << "*" << endl;
}

int main() {
    Murales M;

    inizializzaMurales(M, 3);
    stampaMurales(M);

    coloraCasella(M, 'R', 1, 0);
    coloraCasella(M, 'B', 0, 2);
    coloraCasella(M, 'G', 2, 3);
    stampaMurales(M);

    coloraPrimaCasella(M, 'R');
    stampaMurales(M);

    riempiMurales(M, 'G');

    stampaMurales(M);

    elem *testa = listaPosizioni(M, 'R');
    stampaLista(testa);

    cout << "stringa f2: " << f2("Ciao", "Mondo") << endl;

    f3(6);
    return 0;
}

/* DOMANDA 1: il numero è convertibile,
   * la conversione è pari a 1010111 */

```