```cpp
#include "functions.h"

void initialize(int *Nx, int *Nt, double *T){
    cout << "Nx: "; cin >> *Nx;
    cout << "Nt: "; cin >> *Nt;
    cout << "T : "; cin >> *T ;
}


void ForwardEuler(mat *U, int Nx, int Nt, double h, double dt){
    mat u, unew; double a; ofstream myfile;
    u = *U;
    unew = u;
    a = dt/h/h;

    myfile.open("Explicit_movie.txt");
    for (int t = 0; t<= Nt; t++){
        for (int i = 1; i < Nx-1; i++){
            for (int j = 1; j < Nx-1; j++){
                unew(i,j) = u(i,j) + a*( u(i+1,j) + u(i-1,j) + u(i,j+1) + u(i,j-1) -
4*u(i,j));
            }
        }
    for (int i=0;i<Nx;i++)for(int j=0;j<Nx;j++){{myfile << u(i,j) << " ";}} myfile <<
endl;
    u = unew;
    }
    *U = u;
}


void BackwardEuler(mat *U, int Nx, double h, double dt){
    mat unew, u; double a,a2,difference,tolerance; int n, max_iter; ofstream myfile;
    u = *U;
    a = dt/h/h; a2 = 1 / (1 + 4*a);
    n = 0; difference = 1; tolerance = 1e-6; max_iter = 1e6;

    myfile.open("Implicit_movie.txt");
    while (n < max_iter && difference > tolerance){
        unew = u;
        for (int i=0;i<Nx;i++)for(int j=0;j<Nx;j++){{myfile << u(i,j) << " ";}} myfile
<< endl;
        difference = 0;
        for (int i=1; i<Nx-1; i++){
            for (int j=1; j<Nx-1; j++){
                unew(i,j) = a2*(u(i,j) + a*(unew(i+1,j) + unew(i-1,j) + unew(i,j+1) +
unew(i,j-1)));
                difference += abs(unew(i,j)-u(i,j));
            }
        }
        difference = difference / Nx/Nx;
        n ++;
        u = unew;
    }
    *U = u;
}


void analytical(mat *U){
    mat u; double pi = 3.14;
    u = *U;
    int Nx = u.n_rows; int Ny = u.n_cols;
    double dx = 1./Nx; double dy = 1./Ny;

    for (int i=0; i<Nx; i++){
        for (int j=0; j<Ny; j++){
            for (int n=1; n<100; n++){
                u(i,j) += 2./(n*pi)*(1-cos(n*pi)) * sin(n*pi*j*dy) * ( cosh(n*pi*i*dx) -
cosh(n*pi) / sinh(n*pi) * sinh(n*pi*i*dx));
```

```
            }
        }
    }
    *U = u;
}
```