```cpp
#include <iostream>
#include <cmath>
#include <algorithm>

using namespace std;

typedef long double Num;
typedef vector<vector<Num>> Matriz;
typedef vector<Num> Lista;

void mostrarMatriz(Matriz &A){
    for(int i = 0; i < A.size(); i++){
        for(int j = 0; j < A[i].size(); j++){
            cout<<A[i][j]<<" ";
        }
        cout<<endl;
    }
    cout<<endl;
}

int buscarMayor(int ini, int fin, int j, Matriz &A){
    int mayor = abs(A[ini][j]);
    int index = ini;
    for(int i = ini +1; i <= fin; i++){
        if(mayor < abs(A[i][j])){
            mayor = abs(A[i][j]);
            index = i;
        }
    }
    return index;
}

Lista SustitucionRegresiva(Matriz &A){
    int j = 0;
    Lista res(A.size());
    for(int i = A.size() - 1; i >= 0; i--, j++){
        Num sum = 0;
        for(int k = 0; k < j; k++){
            sum += A[i][A[i].size() - 2 - k] * res[A[i].size() -2 -k];
        }
        res[A[i].size() -2-j] = (A[i][A[i].size()-1] - sum) / A[i][A[i].size()-2-j];

    }
    return res;
}

Matriz Gauss(Matriz A, Lista &B){
    for(int i = 0; i < B.size(); i++){
        A[i].push_back(B[i]);
    }
    int pibot = 0;
    while(pibot != B.size() - 1){
        int mayor = buscarMayor(pibot,B.size()-1,pibot,A);
        if (mayor != pibot) swap(A[pibot],A[mayor]);
        for(int i = pibot+1; i < B.size(); i++){
            Num d = A[i][pibot]/A[pibot][pibot];
            for(int j = pibot; j < A[i].size(); j++){
                A[i][j] = A[i][j] - d*A[pibot][j];
            }
        }
        pibot++;
    }
    return A;
}

int main(){
    Matriz A = {{1,1,-1},{2,-1,1},{4,1,-2}};
    Lista B = {1,2,3};
    Matriz AA = Gauss(A,B);
    Lista res = SustitucionRegresiva(AA);
    for(Num n : res){
        cout<<n<<" ";
    }
    cout<<endl;
}
```