```cpp
#include "PrintModule.h"

static string _PrintModule = RegisterPlugin("Print", new PrintModule());

PrintModule::PrintModule() {}

CASP_Return* PrintModule::Execute(Markup* markup, LanguageDescriptorObject* source_ldo, vector<arg> fnArgs, CASP_Return* inputReturn) {
    returnData = (inputReturn != NULL ? inputReturn : new CASP_Return());

    markup->Print();

    Tree* outputTree = GenerateTree(markup);
    returnData->Data()->Add("ParseTree", outputTree->Output());

    return returnData;

}

Tree* PrintModule::GenerateTree(Markup* m) {
    vector<Markup*> children = m->Children();
    Tree* t = new Tree();
    t->Title = m->GetID();

    if (children.size() > 0) {
        for (int i = 0; i < children.size(); i++) {
            t->Children.push_back(GenerateTree(children[i]));
        }
    } else {
        t->Data = m->GetData();
    }

    return t;
}


GenericObject* Tree::Output() {

    GenericArray* children = new GenericArray();

    for (int i = 0; i < Children.size(); i++) {
        children->Add(Children[i]->Output());
    }

    return new GenericObject({
        {"Data", CreateLeaf(Data)},
        {"Title", CreateLeaf(Title)},
        {"Children", children}
    });

}
```