

```
#include "IniParser.h"

// Charge le contenu du fichier .ini dans une hashtable
IniParser::IniParser(const char* path)
{
    FILE *f = fopen(path, "r");
    char *buffer;
    size_t len = 0;

    if (f == NULL)
        throw Exception("Impossible d'ouvrir le fichier");

    while (getline(&buffer, &len, f) != -1) {
        char *reentrant_buffer, *key, *value;

        key = strtok_r(buffer, "=", &reentrant_buffer);
        value = strtok_r(NULL, "=", &reentrant_buffer);

        // Supprime le \n final
        value[strlen(value) - 1] = '\0';

        this->_assoc.insert(pair<string, string>(string(key), string(value)));
    }

    delete buffer;
    fclose(f);
}

IniParser::IniParser(const IniParser& other)
{
    this->_assoc = other.get_assoc();
}

IniParser& IniParser::operator=(const IniParser& other)
{
    return *this;
}

string IniParser::get_value(const string key) const
{
    map<string, string>::const_iterator value = this->_assoc.find(key);

    if (value == this->_assoc.end())
        throw Exception("La clé n'existe pas");
    else
        return value->second;
}
```