

```

1  #include "parser.h"
2
3  /* FUNCTIONS */
4
5  // fonction Parseur
6  std::vector<elmt_sequence> parser(){
7
8      int pos_separateur; // position du séparateur dans la ligne
9      string maLigne; // temp sur la ligne courante (en lecture)
10     std::vector<elmt_sequence> maSequence; // temp sur la séquence à retourner
11
12     ifstream fichier("./txt/sequence.txt", ios::in); // ouverture du fichier
13
14     // si l'ouverture a réussi
15     if(fichier)
16     {
17         // tant qu'on est pas au bout du fichier
18         while(getline(fichier, maLigne))
19         {
20             // cherche la position du ";" dans la ligne
21             pos_separateur = maLigne.find(';');
22
23             // envoie les valeurs dans la séquence
24             maSequence.push_back(elmt_sequence(maLigne.substr(1,pos_separateur-1),std::stoi((maLigne.
25             substr(pos_separateur+1,maLigne.length()-pos_separateur-2)))));
26         }
27
28         // instructions
29         fichier.close(); // je referme le fichier
30
31     } else {
32
33         // erreur
34         cout << "Erreur à l'ouverture !" << endl;
35
36     }
37
38     return maSequence;
39 }
40
41 // Affiche la séquence comme elle apparait dans le fichier sequence.txt
42 void afficheur_sequence(std::vector<elmt_sequence> seq) {
43
44     vector<elmt_sequence>::iterator it_seq; // itérateur
45     for (it_seq = seq.begin(); it_seq != seq.end() ; it_seq++)
46     {
47         cout<<"("<<(*it_seq).get_label()<<" "<<(*it_seq).get_date()<<" "<<endl;
48     }
49 }
50
51 }
52
53

```