

```

1  #include "parser_chr.h"
2
3  /* FUNCTIONS */
4
5  // fonction Parseur chroniques
6  std::list<chronique> parser_chroniques(){
7
8      string maLigne; // temp sur la ligne courante (en lecture)
9      string label; // temp sur le nom de la chronique
10     int nb_contraintes; // temp sur le nb de contraintes
11
12     int contr_date; // temp sur le contenu d'une contrainte
13     int contr_type; // idem
14     string contr_nom_event; // idem
15     vector<contrainte> vecteur_c; // vecteur temp sur les vecteurs de contrainte
16
17     std::list<chronique> maListe; // temp sur la liste à retourner
18
19     int pos_separateur_1; // position du séparateur ; dans la ligne
20     int pos_separateur_2; // position du séparateur : dans la ligne
21     int pos_par_1; // position de la parenthese ( dans la ligne
22     int pos_par_2; // position de la parenthese ) dans la ligne
23     int pos_virg_1; // position de la premiere virgule dans la parenthese courante
24     int pos_virg_2; // position de la seconde virgule dans la parenthese courante
25
26     ifstream fichier("./txt/chroniques.txt", ios::in); // ouverture du fichier
27
28     // si l'ouverture a réussi
29     if(fichier)
30     {
31         // tant qu'on est pas au bout du fichier
32         while(getline(fichier, maLigne))
33         {
34
35             // cherche la position du ";" dans la ligne
36             pos_separateur_1 = maLigne.find(';');
37
38             // cherche la position du ":" dans la ligne
39             pos_separateur_2 = maLigne.find(':');
40
41             label=maLigne.substr(0,pos_separateur_1);
42             nb_contraintes =
std::stoi((maLigne.substr(pos_separateur_1+1,pos_separateur_2-pos_separateur_1-1)));
43
44
45             // identification des contraintes
46             pos_par_1 = pos_separateur_2+1; // premiere parenthese
47
48             for (int j=0;j<nb_contraintes;j++){
49
50                 // recherche des séparateurs
51                 pos_virg_1 = maLigne.find_first_of(', ',pos_par_1);
52                 pos_virg_2 = maLigne.find_first_of(', ',pos_virg_1+1);
53                 pos_par_2 = maLigne.find_first_of(')',pos_par_1);
54
55                 // affectation des données de la contrainte
56                 contr_nom_event = maLigne.substr(pos_par_1+1,pos_virg_1-pos_par_1-1);
57                 contr_date = std::stoi((maLigne.substr(pos_virg_1+1,pos_virg_2-pos_virg_1-1)));
58                 contr_type = std::stoi((maLigne.substr(pos_virg_2+1,pos_par_2-pos_virg_2-1)));
59
60                 // ajout de la contrainte dans le vecteur contrainte
61                 vecteur_c.push_back(contrainte(contr_date,contr_type,contr_nom_event));
62                 pos_par_1=pos_par_2+1; // nouvelle parenthese
63             }
64
65             // envoie les valeurs dans la liste

```

```

66         maListe.push_back(chronique(label, nb_contraintes, vecteur_c));
67         vecteur_c.clear();
68     }
69
70     // instructions
71     fichier.close(); // je referme le fichier
72
73 } else {
74
75     // erreur
76     cout << "Erreur à l'ouverture !" << endl;
77
78 }
79
80 return maListe;
81
82 }
83
84
85 // Affiche la liste des chroniques comme elle apparait dans le fichier chroniques.txt
86 void afficheur_liste_chr(std::list<chronique> l_chr) {
87
88     list<chronique>::iterator it_chr; // itérateur sur la liste
89     for (it_chr = l_chr.begin(); it_chr != l_chr.end() ; it_chr++)
90     {
91         cout<<(*it_chr).get_nom()<<" "<<(*it_chr).get_contraintes_total()<<":";
92
93         vector<contrainte> vecteur_c = (*it_chr).get_contraintes(); // chargement du vecteur de contrainte
94         vector<contrainte>::iterator it_cont; // itérateur sur la contrainte
95         for (it_cont = vecteur_c.begin(); it_cont != vecteur_c.end() ; it_cont++) {
96
97             cout<<" ("<<(*it_cont).get_event().get_label()<<" "<<(*it_cont).get_time()<<" "<<(*it_cont).get_type()<<")";
98             cout<<endl;
99         }
100     }

```