

```

1  /* #####*/
2  . * » » » » » » » » main.c
3  . *
4  . * Fichier contenant la fonction principal ainsi que Les fonctions de test
5  . *
6  . * #####*/
7
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include "actions.h"
11 #include "semaine.h"
12
13
14 /*-----
15 . * » » » » » afficheStructure
16 . *
17 . * Fonction permettant d'afficher L'état actuel de la structure
18 . * Permettra de vérifier Le résultat d'un jeux de test
19 . *
20 . * Entrée
21 . * » » semaines : structure à afficher
22 . * -----*/
23 void afficheStructure(semaines_t * semaines){
24     » semaines_t * ptrSemaine = semaines; /* variable de parcours de la liste
    • des semaines*/
25     » actions_t » * ptrAction; » » » » » /* variable de parcours des listes
    • d'actions */
26     » while(ptrSemaine != NULL){
27     » » printf("%s :\n", ptrSemaine->semaine );
28     » » ptrAction = ptrSemaine->actions;
29     » » while(ptrAction != NULL){
30     » » » printf("    %s %s\n", ptrAction->jour_heure, ptrAction->nom_action);
31     » » » ptrAction = ptrAction->suiv;
32     » » }
33     » ptrSemaine = ptrSemaine->semaineSuivante;
34     » }
35 }
36
37 /* -----
38 . * » » » » » testFonctionLireFichier
39 . *
40 . * Fonction qui test la Lecture d'un fichier dans Le cas normale
41 . * ----- */
42 void testFonctionLireFichier(){
43     » semaines_t * structure = (semaines_t *) malloc(sizeof(semaines_t));
44     » structure = lireFichier("liste_agenda"); /* élément récupérant
45     » la structure après Lecture */
46     » afficheStructure(structure);
47 }
48
49 /* -----
50 . * » » » » » testFonctionLireMauvaisNomFichier
51 . *
52 . * Fonction qui test la Lecture d'un fichier dans Le cas d'un mauvais
53 . * nom de fichier

```

```

54  /* ----- */
55  void testFonctionLireMauvaisNomFichier(){
56  » semaines_t * structure = (semaines_t *) malloc(sizeof(semaines_t));
57  » structure = lireFichier("nom_errone"); /* élément récupérant
58  » la structure après Lecture (ici Lecture impossible) */
59  » afficheStructure(structure);
60  }
61
62  /* -----
63  » » » » testCreateSemaine
64  »
65  » Fonction qui test La création d'une semaine dans La structure dans
66  » Le cas normale
67  » ----- */
68  void testCreateSemaine(){
69  » semaines_t * structure = (semaines_t *) malloc(sizeof(semaines_t));
70  » /*Structure de test */
71  » createSemaine(structure, "201742413Partiels");
72  » afficheStructure(structure);
73  }
74
75  /* -----
76  » » » » testCreateSemaineDoublon
77  »
78  » Fonction qui test La création d'une semaine dans La structure dans
79  » Le cas où La plage horaire n'est pas disponible
80  » ----- */
81  void testCreateSemaineDoublon(){
82  » semaines_t * structure = (semaines_t *) malloc(sizeof(semaines_t));
83  » /*Structure de test */
84  » createSemaine(structure, "201742413Partiels");
85  » createSemaine(structure, "201742413Medecin");
86  » afficheStructure(structure);
87  }
88
89  /* -----
90  » » » » testCreateSemaineMemeJour
91  »
92  » Fonction qui test La création d'une semaine dans La structure dans
93  » Le cas où il y a déjà une action La même semaine et Le même jour
94  » ----- */
95  void testCreateSemaineMemeJour(){
96  » semaines_t * structure = (semaines_t *) malloc(sizeof(semaines_t));
97  » /*Structure de test */
98  » createSemaine(structure, "201742413Partiels");
99  » createSemaine(structure, "201742416Medecin");
100 » afficheStructure(structure);
101 }
102
103 /* -----
104 » » » » testSauvegardeSemaine
105 »
106 » Fonction qui test La sauvegarde de La structure dans un fichier
107 » ----- */
108 void testSauvegardeSemaine(){

```

```

109  » semaines_t * structure = (semaines_t *) malloc(sizeof(semaines_t));
110  » /*Structure de test */
111  » createSemaine(structure, "201742413Partiels");
112  » createSemaine(structure, "201732209Voyage Pau");
113  » createSemaine(structure, "201742416Restaurant");
114  » sauvegardeSemaine(structure, "testSauvegarde.txt");
115  » afficheStructure(structure);
116  }
117
118  /* -----
119  * » » » » testSauvegardeSemaineMauvaisNomFichier
120  *
121  * Fonction qui test La sauvegarde de La structure dans un fichier avec
122  * un mauvais nom
123  * -----*/
124  void testSauvegardeSemaineMauvaisNomFichier(){
125  » semaines_t * structure = (semaines_t *) malloc(sizeof(semaines_t));
126  » /*Structure de test */
127  » createSemaine(structure, "201742413Partiels");
128  » createSemaine(structure, "201732209Voyage Pau");
129  » createSemaine(structure, "201742416Restaurant");
130  » sauvegardeSemaine(structure, "nom_errone.txt");
131  » afficheStructure(structure);
132  }
133
134  /* -----
135  * » » » » testCreateListeJourFromActionsSemaine
136  *
137  * Fonction qui test La création d'une liste de jours suivant une chaine
138  * passée en paramètre
139  * -----*/
140  void testCreateListeJourFromActionsSemaine(){
141  » semaines_t * structure = (semaines_t *) malloc(sizeof(semaines_t));
142  » /*Structure de test */
143  » createSemaine(structure, "201742413Partiels");
144  » createSemaine(structure, "201732209Partiels");
145  » createSemaine(structure, "201742416Restaurant");
146  » printf("%s\n\n", createListeJourFromActionsSemaine(structure, "Partiels"));
147  » afficheStructure(structure);
148  }
149
150  /* -----
151  * » » » » testCreateListeJourFromActionsSemaineSansJours
152  *
153  * Fonction qui test La création d'une liste de jours suivant une chaine
154  * passée en paramètre dans le cas où il n'y a pas d'action correspondante
155  * -----*/
156  void testCreateListeJourFromActionsSemaineSansJours(){
157  » semaines_t * structure = (semaines_t *) malloc(sizeof(semaines_t));
158  » /*Structure de test */
159  » createSemaine(structure, "201742413Partiels");
160  » createSemaine(structure, "201732209Partiels");
161  » createSemaine(structure, "201742416Restaurant");
162  » printf("%s\n\n", createListeJourFromActionsSemaine(structure, "Medecine"));
163  » afficheStructure(structure);

```

```

164 }
165
166 /* -----
167  * » » » » » testSupprimeActionInSemaines
168  *
169  * Fonction qui test La suppression d'une action à partir de sa date et
170  * son heure
171  * ----- */
172 void testSupprimeActionInSemaines(){
173     » semaines_t * structure = (semaines_t *) malloc(sizeof(semaines_t));
174     » /*Structure de test */
175     » createSemaine(structure, "201742413Partiels");
176     » createSemaine(structure, "201732209Partiels");
177     » createSemaine(structure, "201742416Restaurant");
178     » supprimeActionInSemaines(structure, "201732209");
179     » afficheStructure(structure);
180 }
181
182 /* -----
183  * » » » » » testSupprimeActionInSemainesSansActions
184  *
185  * Fonction qui test La suppression d'une action à partir de sa date et
186  * son heure dans le cas où il n'y a rien à supprimer
187  * ----- */
188 void testSupprimeActionInSemainesSansActions(){
189     » semaines_t * structure = (semaines_t *) malloc(sizeof(semaines_t));
190     » /*Structure de test */
191     » createSemaine(structure, "201742413Partiels");
192     » createSemaine(structure, "201732209Partiels");
193     » createSemaine(structure, "201742416Restaurant");
194     » supprimeActionInSemaines(structure, "201932210");
195     » afficheStructure(structure);
196 }
197
198
199 int main(){/*Décommenter la fonction à tester */
200
201     » /* testFonctionLireFichier(); */
202     » /* testCreateSemaine(); */
203     » /* testCreateSemaineDoublon(); */
204     » /* testCreateSemaineMemeJour(); */
205     » /* testSauvegardeSemaine(); */
206     » /* testSauvegardeSemaineMauvaisNomFichier(); */
207     » /* testCreateListeJourFromActionsSemaine(); */
208     » /* testCreateListeJourFromActionsSemaineSansJours(); */
209     » /* testSupprimeActionInSemaines(); */
210     » /* testSupprimeActionInSemainesSansActions(); */
211
212     » return 0;
213 }

```