

```

1  /*-----
•  --*/
2  /*» actions.h»» fichier contenant la déclaration de la structure*/
3  /*» » » de la liste chaînée qui les actions et la » » */
4  /*» » » déclaration des fonctions utilisées»» » */
5  /*-----
•  --*/
6
7  #ifndef __ACTIONS_H_GUILLAUME_FARAH__
8  #define __ACTIONS_H_GUILLAUME_FARAH__
9
10 /*#include "semaine.h"
11 #include "main.c"*/
12 #include <string.h>
13 #include <stdio.h>
14 #include <stdlib.h>
15
16
17
18 typedef struct actions{
19     ...char jour_heure [3];
20     ...char nom_action [10];
21     ...struct actions * suiv;
22 } actions_t;
23
24
25
26
27
28 /*-----
•  --*/
29 /*CreateAction» » Insérer une nouvelle action à la suite de L'agenda» */
30 /*» » » » » » » » */
31 /*En sortie:» » La fonction retourne une nouvelle liste chaînée contenant
32 /*» » une nouvelle action si la plage horaire n'est pas déjà occupée» */
33 /*-----
•  --*/
34
35 void CreateAction(actions_t *, char *);
36
37
38
39
40
41 /*-----
•  --*/
42 /*AlloueAction» » Allouer de l'espace mémoire à la suite du pointeur » */
43 /*» » » Insérer le nouvel élément dans cet espace mémoire*/
44 /*» » » » » » » » */
45 /*En sortie:» » La fonction retourne une nouvelle liste chaînée contenant
46 /*» » un nouvel espace mémoire et une nouvelle action» » */
47 /*-----
•  --*/
48
49 void AlloueAction( actions_t *, char [3], char [10]);

```



```

50  ↵
51  ↵
52  ↵
53  ↵
54  ↵
55  /*-----
   •  --*/↵
56  /*Recherche» Recherche une valeur dans une liste chaînée» */↵
57  /*» » » » » » » » */↵
58  /*En sortie:» La fonction retourne l'adresse de l'élément précédent »
59  /*» » » où il faut insérer la nouvelle valeur» */↵
60  /*-----
   •  --*/↵
61  ↵
62  actions_t ** Recherche ( actions_t *, char [3]);↵
63  ↵
64  ↵
65  ↵
66  ↵
67  ↵
68  /*-----
   •  --*/↵
69  /*Sauvegarde» fonction qui sauvegarde la SDD sous la forme d'un fichier
70  /*» » » » » » » » */↵
71  /*En sortie:» Renvoie un fichier contenant une sauvegarde de la SDD*/↵
72  /*-----
   •  --*/↵
73  ↵
74  void Sauvegarde ( FILE *, actions_t *, char *);↵
75  ↵
76  ↵
77  ↵
78  ↵
79  ↵
80  /*-----
   •  --*/↵
81  /*Sauvegarde» Fonction qui cherche si une action est présente dans */↵
82  /*» » » l'agenda» » » » */↵
83  /*» » » » » » » » */↵
84  /*En sortie:» Renvoie, si l'action est présente, les jours qui */↵
85  /*» » contiennent une ou plusieurs fois cette action sous forme de */↵
86  /*» » tableau de caractères» » » » */↵
87  /*-----
   •  --*/↵
88  ↵
89  char * CreateListeFromActions ( actions_t *, char *);↵
90  ↵
91  ↵
92  ↵
93  ↵
94  ↵
95  /*-----
   •  --*/↵
96  /*SupprimeAction» Fonction qui supprime une action en connaissant */↵
97  /*» » » la date et l'heure» » » » */↵

```

```
98  /*» » » » » » » » */↵
99  /*En sortie:» » la fonction retourne une nouvelle liste chaînée ne */↵
100 /*» » contenant plus l'action à supprimer si elle existe» » */↵
101 /*-----↵
   • --*/↵
102 void SupprimeAction ( actions_t *, char [3]);↵
103 ↵
104 ↵
105 ↵
106 ↵
```