Rapport du premier projet d'IAP

Groupe 101 et 103

Noé GODIN Louis MASSON

Table des matières

1) Présentation de l'application	page 3
----------------------------------	--------

- 2) Organisation des tests page 4
- 3) Bilan de validation des tests de développement page 5-8
- 4) Bilan du projet page 9
- 5) Code C complet (annexe s) page 10-13

1) Présentation de l'application

Cette application a pour objectif de mémoriser et de manipuler jusqu'à dix tournois de tennis féminin. Il fonctionne grâce à des lignes de commande en langage C et peut :

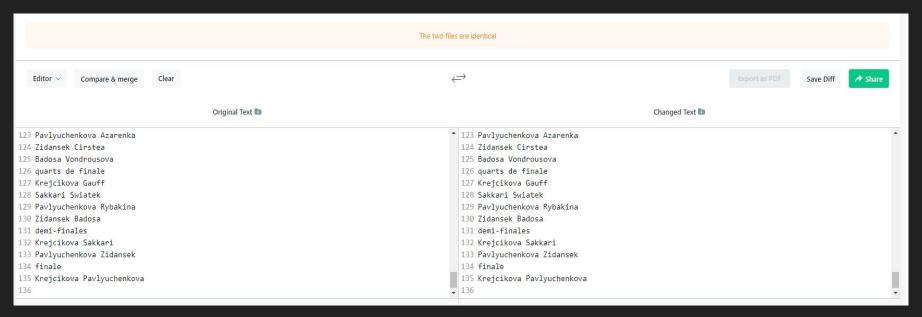
- enregistrer les joueuses participant à des tournois donnés
- afficher le classement d'un ou plusieurs tournois et le nombre de points gagnés par chaque joueuse
- rechercher les matchs précis d'une joueuse particulière dans un tournoi particulier

2) Organisation des tests

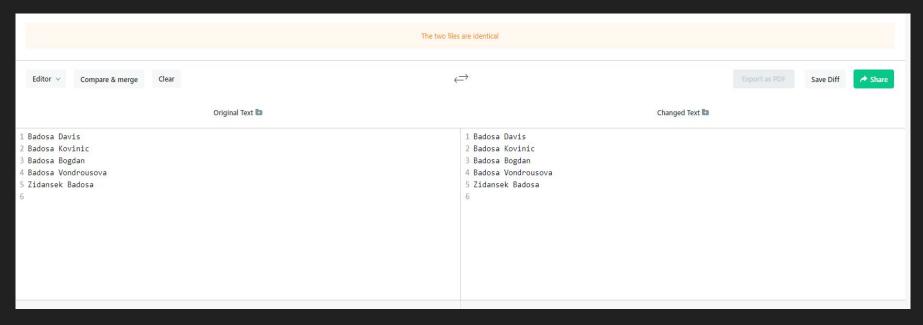
Afin de vérifier la validité de nos sprints, nous avons testé chacun d'entre eux avec des JDT et par la suite nous avons comparé nos résultats aux résultats attendus à l'aide du site diffchecker.

Ce site met en évidence les différences entre deux fichiers texte. Ainsi nous avons pu vérifier que notre application fonctionne correctement.

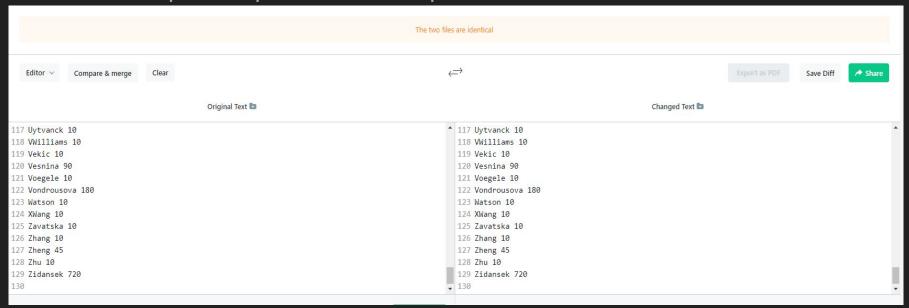
Ce test vérifie si on peut enregistrer et afficher un tournoi précis. Il valide le Sprint 1.



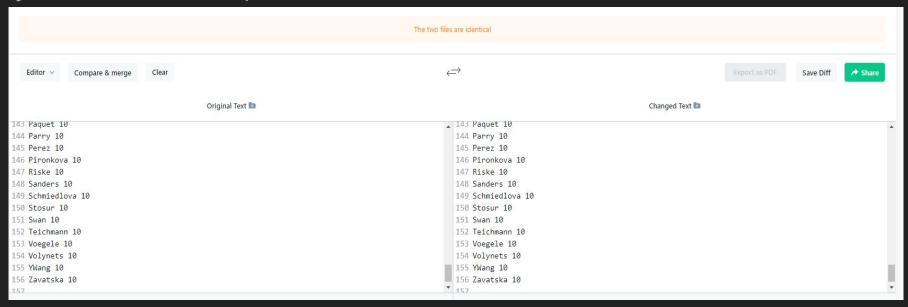
Ce test vérifie si on peut afficher une joueuse d'un tournoi précis et ses matchs. Il valide le Sprint 2.



Ce test vérifie si on peut afficher un tournoi précis et le score de chaque joueuse dans l'ordre alphabétique. Il valide le Sprint 3.



Ce test vérifie si on peut afficher chaque tournoi et le cumul de points de chaque joueuse. Il valide le Sprint 4.



4) Bilan de projet

Ce projet nous a fait énormément progresser en C.

Il nous a permis d'approfondir nos connaissances des types structurés, des tableaux et des pointeurs, et d'utiliser ces types de données.

Il a également mis en évidence l'importance des fonctions dans un programme, permettant l'organisation et l'optimisation du code.

Nous avons eu des difficultés pour comprendre le sprint 5 et nous n'avons pas eu assez de temps ensemble pour le terminer. Notre programme pourrait être amélioré si l'on modifiait la fonction "afficher_classement" de façon à valider le Sprint 5.

En conclusion, cet exercice nous a montré à quel point il faut être attentif à chaque étape de la conceptualisation du code, du fonctionnement et des sorties de programme.

5) Codes source

Les codes source sont disponibles dans le dossier Annexe.

Le dernier fichier (source.c) est entièrement commenté.

Code C I.1 à I.58

```
//sprint 1 à 4
     #include <stdlib.h>
     #include <stdio.h>
     #include <time.h>
     #include <string.h>
     #pragma warning (disable : 4996)
 8
     #define maxTournois 10
 9
     #define nbMatchTournoi 127
10
     #define nbJoueusesTournoi 128
11
     #define longueurMot 30
12
13
     typedef struct {
14
         char nomflongueurMot + 11:
15
         unsigned int points:
     }Joueuse:
17
18
     typedef struct {
19
         unsigned int idxGagnante:
20
         unsigned int idxPerdante:
21
     Match:
22
23
     typedef struct {
24
         char nom[longueurMot + 1];
25
26
         Match dataMatch[nbMatchTournoil:
27
     }Tournoi:
28
29
     typedef struct (
30
        Tournoi dataTournois[maxTournois]:
31
         Joueuse dataJoueuse[maxTournois * nbJoueusesTournoi];
32
         unsigned int nbTournois;
33
         unsigned int idxJ;
34
         unsigned int idxT:
35
     }TournoiWTA:
36
37
     //prototypage des fonctions utilisees et documentation
38
39
     * demande le nombre de tournoi avec lequel on travaille
41
     * [in] tableau de type tournoiWTA
42
43
     void definir_nombre_tournois(TournoiWTA* ins);
44
45
     * regarde si le tournoi recherché existe
     * [in] tableau de type tournoiWTA
     * [in] tableau de type char
     * [in] entier
50
     * [retour] entier
51
     int ExistanceTournoi(const TournoiWTA* ins, const char ville[], const int date);
53
54
     * enregistre un tournoi spécifié
56
     * [in] tableau de type tournoiWTA
57
     void enregistrement_tournoi(TournoiWTA* ins);
59
```

Code C I.60 à 116

```
61 * inscrit ou retrouve l'index d'une joueuse
 62 * [in] tableau de type tournoiWTA
 63 * [in] tableau de type char (nom de la joueuse)
 64 * [retour] entier (index joueuse)
66 int referenceID(TournoiWTA* ins, char joueuse[])
 69 * Ajoute des points aux joueuses dans le tournoi
 70 * [in] tableau de type tournoiWTA
72 yoid attributionpoints(TournoiWTA* ins):
74 /*
 75 * Affiche un tournoi spécifié
 76 * [in] tableau de type tournoiWTA
78 void affichage_matchs_tournoi(const TournoiWTA* ins);
80 /*
 81 * Affiche les matchs d'une joueuse
 82 * [in] tableau de type tournoiWTA
     void afficher matchs joueuse(const TournoiWTA*);
 87 * Affiche le score de chaque joueuse dans un tournoi donné
     * [in] tableau de type tournoiWTA
 90 void affichage joueuses tournoi(TournoiWTA* ins);
 93 * Affiche le cumul de points de chaque joueuse
 94 * [in] tableau de type tournoiWTA
 95 */
 96 void afficher classement(TournoiWTA* ins);
     void definir nombre tournois(TournoiWTA* ins) {
         int x:
         scanf("%d", &x);
         if ((x >= 1) && (x <= 10)) { //on verifie si il y a bien un tournoi compris entre 1 et 10 compris on ne pourra pas enregistrer moins de 1 tournoi et pas plus de 10
             ins->nbTournois = x; //écrit le nombre de tournoi dans la structure tournoiWTA
103
104
      int ExistanceTournoi(const TournoiWTA* ins, const char ville[], const int date) {
         for (int i = 0; i < ins->idxT; \leftrightarrowi) {
             if ((strcmp(ins->dataTournois[i].nom, ville) == 0) && (ins->dataTournois[i].date == date)) {
110
                 return i: //renvoi le tournois avant le même nom / date que celui demandé sinon renvoi -1
111
112
         printf("tournoi inconnu");
         return -1;
115
```

Code C I.127 à 175

```
void enregistrement tournoi(TournoiWTA* ins) {
          char ville[longueurMot + 1]:
119
          int date;
120
          scanf("%s", ville);
121
          scanf("%d", &date):
122
          strcpy(ins->dataTournois[ins->idxT].nom, ville);
123
          ins->dataTournois[ins->idxT].date = date;
124
125
          char joueuseGagne[longueurMot + 1];
126
          char joueusePerd[longueurMot + 1];
127
128
          for (int countMatchs = 0: countMatchs < nbMatchTournoi: ++countMatchs) {
129
              scanf("%s", joueuseGagne)://Lorsqu'on enregistre, la premiere joueuse (a gauche) est toujours gagnate
130
              scanf("%s", joueusePerd);//A l'invere, la deuxieme joueuse (a droite) est toujours perdante
131
132
              ins->dataTournois[ins->idxT].dataMatch[countMatchs].idxGagnante = referenceID(ins, ioueuseGagne)://on recupere son index
133
              ins->dataTournois[ins->idxT].dataMatch[countMatchs].idxPerdante = referenceID(ins. joueusePerd)://on recupere son index
134
135
          attributionpoints(ins);//on leur attributs leurs points
136
          ins->idxT += 1:
137
138
139
       void attributionpoints(TournoiWTA* ins) {
148
           for (int i = 0; i < nbMatchTournoi; ++i) {//boucle sur tous les matchs pour ajouter des points aux perdants
141
              if (i >= 0 && i < 64) ins->dataJoueuse[ins->dataTournois[ins->idxT].dataMatch[i].idxPerdante].points += 10;
142
              else if (i >= 64 && i < 96) {
143
                  ins->dataJoueuse[ins->dataTournois[ins->idxT].dataMatch[i].idxPerdante].points += 45;
144
145
146
                  ins->dataJoueuse[ins->dataTournois[ins->idxT].dataMatch[i].idxPerdante].points += 90;
147
148
               else if (i >= 112 && i < 120) {
149
                  ins->dataJoueuse[ins->dataTournois[ins->idxT].dataMatch[i].idxPerdante].points += 180;
150
151
               else if (i >= 120 && i < 124) {
152
                  ins->dataJoueuse[ins->dataTournois[ins->idxT].dataMatch[i].idxPerdante].points += 360;
153
154
               else if (i >= 124 && i < 126) {
155
                  ins->dataJoueuse[ins->dataTournois[ins->idxT].dataMatch[i].idxPerdante].points += 720;
156
157
               if (i == 126) {
158
                  ins->dataJoueuse[ins->dataTournois[ins->idxT].dataMatch[i].idxGagnante].points += 2000;//ajout des points a la gagnante
159
                  ins->dataJoueuse[ins->dataTournois[ins->idxT].dataMatch[i].idxPerdante].points += 1200;
160
161
162
163
164
       int referenceID(TournoiWTA* ins, char joueuse[]) {
165
          for (int i = 0: i < ins->idxJ: ++i) { //pour toutes les joueuses déjà inscrite
166
              if (strcmp(ins->dataJoueuse[i].nom, joueuse) == 0) {
167
                  return i; //test qui renvoi i si joueuse est dans le tableau
168
169
170
          strcpy(ins->dataJoueuse[ins->idxJ].nom. joueuse): //si le nom n'est pas dans le tableau, alors on le met dedans
171
          ins->dataJoueuse[ins->idxJ].points = 0; //initialisation des points des nouvelles joueuses à 0
          ins->idx3 += 1;
174
          return ins->idx3 - 1; //renvoi l'index de notre joueuse
175 }
```

Code C I.177 à 234

```
void affichage_matchs_tournoi(const TournoiWTA* ins) {
          char ville[longueurMot + 1];
179
          int date:
180
          scanf("%s", ville);
          scanf("%d", &date);
          int t = ExistanceTournoi(ins, ville, date);
183
          if (t == -1) {
184
             return:
185
186
          printf("%s %d\n", ins->dataTournois[t].nom, ins->dataTournois[t].date);//ecriture du nom du tournoi et de sa date
187
           for (int i = 0; j < nbMatchTournoi; ++j) {
188
             if (i == 0) printf("64emes de finale\n");
189
              else if (i == 64) {
190
                 printf("32emes de finale\n");
191
192
              else if (i == 96) {
193
                 printf("16emes de finale\n");
194
195
              else if (j == 112) {
196
                  printf("8emes de finale\n");
197
198
              else if (j == 120) {
199
                 printf("quarts de finale\n");
200
201
              else if (i == 124) {
202
                  printf("demi-finales\n");
203
204
              else if (i == 126) {
205
                 printf("finale\n");
206
207
              printf("%s ", ins->dataJoueuse[ins->dataTournois[t].dataMatch[j].idxGagnante].nom);//affichage de la gagnante
208
              printf("%s\n", ins->dataJoueuse[ins->dataTournois[t].dataMatch[i].idxPerdante].nom);//affichage de la perdante
209
210
211
      void afficher matchs joueuse(const TournoiWTA* ins) {
          char ville[longueurMot + 1]:
214
          int date, i;
215
          scanf("%s", ville)
          scanf("%d", &date);
217
           int t = ExistanceTournoi(ins, ville, date);
218
          if (t == -1) {
219
              return;
228
          char joueuse[longueurMot + 1];
          scanf("%s", ioueuse);
223
           for (int i = 0; i < ins->idxJ; ++i) {
224
              if (strcmo(joueuse, ins->dataJoueuse[i].nom) == 0) {//on regarde si le nom de la joueuse est presente dans un tournoi.
225
                  for (int i = 0; i < nbMatchTournoi; ++i) {
226
                      if ((i == ins->dataTournois[t].dataMatch[i].idxGagnante) || (i == ins->dataTournois[t].dataMatch[i].idxPerdante)) { //on affiche ses matchs
227
                          printf("%s ", ins->dataJoueusefins->dataTournoisftl.dataMatchfil.idxGagnantel.nom);
228
                          printf("%s\n", ins->dataJoueuse[ins->dataTournois[t].dataMatch[i].idxPerdantel.nom);
229
230
231
232
233 }
```

Code C I.235 à I.292

```
void affichage_joueuses_tournoi(TournoiWTA* ins) {
          char ville[longueurMot + 1];
          int date, i, Retenu1, Retenu2, points[nb]oueusesTournoi], Joueuses[nb]oueusesTournoi];
          int notrejoueuse = 0; //remplace par i
          scanf("%s", ville):
          scanf("%d", &date);
241
          int t = ExistanceTournoi(ins, ville, date);
242
          if (t == -100) {
243
244
245
          printf("%s %d\n", ins->dataTournois[t].nom, ins->dataTournois[t].date);
          for (i = 0; i < 64; i++) { //dans les 64 premiers matchs les perdantes ont 10 points
247
              Joueuses[notrejoueuse] = ins->dataTournois[t].dataMatch[i].idxPerdante;
248
              points[notrejoueuse] = 10:
249
              notreioueuse++:
250
251
          for (; i < 96; i++) { //dans les 32 prochains matchs les perdantes ont 45 points (64 + 32 = 96)
252
              Joueuses[notrejoueuse] = ins->dataTournois[t].dataMatch[i].idxPerdante;
253
              points[notrejoueuse] = 45;
254
              notrejoueuse++;
255
256
          for (: i < 112: i++) { //dans les 16 prochains matchs les perdantes ont 90 points
257
              Joueuses[notrejoueuse] = ins->dataTournois[t].dataMatch[i].idxPerdante;
258
              points[notrejoueuse] = 90;
259
              notreioueuse++:
260
261
          for (; i < 120; i++) { //dans les 8 prochains matchs les perdantes ont 180 points
262
              Joueuses[notrejoueuse] = ins->dataTournois[t].dataMatch[i].idxPerdante;
263
              points[notrejoueuse] = 180;
264
              notreioueuse++:
265
266
          for (; i < 124; i++) { //dans les 4 prochains matchs les perdantes ont 360 points
267
              Joueuses[notrejoueuse] = ins->dataTournois[t].dataMatch[i].idxPerdante;
268
              points[notreioueuse] = 360:
269
              notrejoueuse++;
270
271
          for (; i < 126; i++) { //dans les 2 prochains matchs les perdantes ont 720 points
272
              Joueuses[notrejoueuse] = ins->dataTournois[t].dataMatch[i].idxPerdante;
273
              points[notrejoueuse] = 720;
274
              notrejoueuse++;
275
          Joueuses[notreioueuse] = ins->dataTournois[t].dataMatch[i].idxPerdante:
          points[notrejoueuse] = 1200; //la perdante de la finale à 1200 points
          notrejoueuse++;
          Joueuses[notrejoueuse] = ins->dataTournois[t].dataMatch[i].idxGagnante:
280
          points[notrejoueuse] = 2000; //la gagnante du tournoi à 2000 points
281
          for (int i = 1; i < nbJoueusesTournoi; ++i)
282
              if (strcmp(ins->dataJoueuse[Joueuses[i - 1]].nom, ins->dataJoueuse[Joueuses[i]].nom) > 0) {
283
                  Retenu1 = Joueuses[i - 1];
284
                  Retenu2 = points[i - 1]:
285
                  Joueuses[i - 1] = Joueuses[i]:
286
                  points[i - 1] = points[i];
287
                  Joueuses[i] = Retenu1;
288
                  points[i] = Retenu2;
289
                 i = 0:
          for (int i = 0: i < nbJoueusesTournoi: ++i)
              printf("%s %d\n", ins->dataJoueuse[Joueuses[i]].nom, points[i]);
```

Code C I.293 à I.351

```
295 void afficher_classement(TournoiWTA* ins) {
          if (ins->nbTournois == 0) printf("pas de classement"):
          int tab[maxTournois * nb]oueusesTournoil:
          int i, h, idxGagnante, idxPerdante;
          unsigned int onRetient, j, elemTableau, Valsuperieur;
          for (i = 0: i < ins->idx3: ++i)
              tab[i] = i:
          for (i = 0; i < ins->idxJ; ++i) {
              Valsuperieur = i:
              for (elemTableau = j; elemTableau < ins->idxJ; ++elemTableau)
                  if (ins->data]oueuse[tab[Valsuperieur]].points < ins->data]oueuse[tab[elenTableau]].points) { //si les joueuses ne sont pas trie par leur point de façon decroissante
                      Valsuperieur = elemTableau:
308
                  if (ins->data]oueuse[tab[Valsuperieur]].points == ins->data]oueuse[tab[elemTableau]].points) {
309
                      if (strcmp(ins->data]oueuse[tab[elemTableau]].nom, ins->data]oueuse[tab[Valsuperieur]].nom) < 1) { // si deux joueuses ont le meme nombre de points mais ne sont pas trie en ordre alphabetque
310
                         Valsumerieur = elemTahleaus
312
313
              }//trieur
              onRetient = tab[j];
              tab[i] = tab[Valsuperieur];
316
              tab[Valsuperieur] = onRetient;
317
          for (i = 0; i < ins->idxJ; ++i)
319
              printf("%s %d\n", ins->data]oueuse[tab[i]].nom, ins->dataJoueuse[tab[i]].nom, ins->dataJoueuse[tab[i]].points);//affichage des joueuses par ordre de point décroissant puis alphabétique dans tous les tournois
328
322 int main() {
          TournoiWTA ins: //creation du tournoi generale du nom de ins
          ins.idxJ = 0; //mise en valeur des indices des joueuses
          ins.nbTournois = 0: //mise en valeur du nombre de tournoi
          ins.idxT = 0: //mise en valeur de l'indice des tournois
          while (1) {
              scanf("%s", mot):
              if (strcmp(mot, "exit") == 0) exit(0); //si le mot exit est tape, alors le programme s'arrete
              else if (strcmp(mot, "definir nombre tournois") == 0) {//si le mot definir nombre tournois est tape, alors la fonction de definition du nombre de tournoi s'execute
                  definir nombre tournois(&ins):
333
              else if (strcmo(mot. "enregistrement tournoi") == 0) {//si le mot enregistrement tournoi est tape, alors la fonction d'enregistrement tournoi s'execute
                  enregistrement tournoi(&ins);
              else if (strcmp(mot, "affichage matchs tournoi") == 0) {//si le mot affichage matchs tournoi est tape, alors on affiche les matchs et leurs deroulement
338
                  affichage matchs tournoi(&ins);
              else if (strcmp(mot, "afficher matchs joueuse") == 0) {//si le mot afficher matchs joueuse est tape, alors on affiche tous les matchs d'une joueuse lors d'un tournoi
              else if (strong(mot. "affichage joueuses tournoi") == 0) (//si le mot affichage joueuses tournoi est tape, alors un classement de point sera affiché par ordre alphabetique
344
                  affichage foueuses tournoi(&ins):
              else if (stromp(mot. "afficher classement") == 0) {//si le mot afficher classement est tape, alors on affiche les points par ordre décroissant et leurs noms par ordre alphabétique si leurs points sont esaux
                  afficher classement(&ins);
348
350
         system("pause"); return 0;
```