

	ANNEE UNIVERSITAIRE 2018-2019 SESSION 1 UE : 4TSI402U Epreuve : 4TSI402EXI Examen d'informatique modulaire Date : 6 mai 2019 Heure : 14h30 Durée : 1h30 Tout document papier autorisé Épreuve de M. Marc
---	--

Gestion des points de plusieurs parties de tennis simultanées

Le programme qui suit est le début d'une étude pour un système centralisé d'affichage des scores en temps réel pour un tournoi de tennis. La version présentée ici est une version basique où seules quelques opérations ont été programmées. Il est conçu de façon à être facilement extensible. Les fonctionnalités implémentées sont prévues pour pouvoir gérer en mode terminal (clavier + écran texte) plusieurs parties de tennis. Il manque quelques améliorations pour que le programme soit utilisable en l'état.

En annexe 1 sont rappelées les règles de décompte des points au tennis.

Un extrait d'une session incluant le début d'une exécution se trouve en annexe 2. Les commandes entrées par l'utilisateur sont en caractères gras.

Le listing de certaines parties des fichiers source se trouvent en annexe 3. Les numéros de ligne ne font pas partie du fichier mais ont été ajoutés à l'impression pour en faciliter le repérage.

Conseils : Prendre le temps de lire tout le sujet, y compris les annexes, avant de commencer à répondre aux questions. Ne pas lire tout le fichier source tout de suite mais commencer par les en-têtes.

N.B. : Les quatre premières questions ne nécessitent que quelques lignes de réponse. Pour les autres, on indiquera uniquement les modifications.

Les questions sont toutes indépendantes.

Le code est constitué de 5 fichiers : `match.c`, `match.h`, `tournoi.c`, `tournoi.h`, `tennis.c`. Une partie du code du programme est donnée en annexe. Certaines fonctions sont incomplètes. Les compléter fera l'objet du problème.

- 1) (3 points) Écrivez un `makefile` permettant de compiler le code.

Si vous ne savez pas l'écrire, indiquez quelle(s) ligne(s) de commande il faut taper dans un terminal pour compiler le programme (en admettant qu'il a été convenablement complété). On pourra utiliser une seule ou plusieurs lignes de commande pour la compilation.

- 2) (1 point) Dans la fonction `tournoiEstTermine()`, `tournoi.c` ligne 22, que signifient les symboles `&&` ? Donnez des exemples significatifs.
- 3) (1 point) Dans `tennis.c`, ligne 11, que signifie le symbole `&` dans l'expression `init_tournoi(&le_tournoi)`.
- 4) (1,5 point) Dans `match.c`, ligne 38, quels sont (en français) les types respectifs de :

```

jeu
jeu->balles_gagnes
jeu->balles_gagnes[0]
```

- 5) (3 points) Modifiez la fonction `main` pour qu'en cas d'erreur flagrante de saisie du numéro du joueur gagnant la balle ou du numéro du match, il n'y ait pas de « plantage » du programme et que l'utilisateur du programme puisse saisir à nouveau le joueur gagnant.
- 6) (5 points) Il manque le fichier `tournoi.h`. Écrivez son code complet (en vous appuyant sur le code de `tournoi.c`).
- 7) (2 points) Il manque la fonction `init_set`. Écrivez sa déclaration et sa définition en indiquant dans quels fichiers vous les écrivez.

- 8) (4 points) La version ci-après du programme ne gère pas la règle de l'avantage (cf annexe 1 et 2) : le premier joueur qui dépasse 40 gagne le jeu. Modifiez le programme pour que la règle de l'avantage soit correctement jouée et affichée conformément aux usages.

1 ANNEXE 1 : Règle de décompte des points au tennis

On ne tient pas compte ici de la règle optionnelle des « Tie Break ».

Un match se déroule en plusieurs sets. Le match est gagné par le premier joueur qui a gagné 2 ou 3 sets (selon les catégories de joueurs) ?

Un set est composé de jeux. Un set est gagné par le joueur qui gagne au moins 6 jeux avec au moins 2 jeux d'avance sur son adversaire.

Un jeu se déroule en plusieurs balles. Une balle gagnée augmente le score du joueur qui l'a gagnée en suivant la suite¹ : 0, 15, 30, 40. Pour gagner le jeu, un joueur doit dépasser 40 et avoir gagné au moins deux balles de plus que son adversaire. Le comptage officiel se fait de la façon suivante. Lorsqu'un joueur a un score de 40 et qu'il gagne une balle, il gagne le jeu si son adversaire a moins de 40. Si l'adversaire avait lui aussi un score de 40, on joue la règle de l'avantage : le joueur qui a gagné le point prend l'« avantage ». S'il gagne le point suivant, il gagne le jeu. Si c'est son adversaire, l'avantage est perdu et il y a « égalité ». Ensuite, le joueur qui gagne une balle sur une égalité prend l'avantage (et on revient au cas précédent)

2 ANNEXE 2 : exemple de session

```
Combien de matchs simultanés ? 2
nom du premier joueur ? John
nom du second joueur ? Yannick
nom du premier joueur ? Roger
nom du second joueur ? Rafael
Pour indiquer un point gagné :
indiquez le numéro de match (1 à 2)
et le numéro du joueur qui marque (1 à 2)
Point gagné par : (match joueur) ? 1 2
0 a 15
Point gagné par : (match joueur) ? 2 1
15 a 0
Point gagné par : (match joueur) ? 2 2
15 a 15
Point gagné par : (match joueur) ? 1 1
15 a 15
Point gagné par : (match joueur) ? 1 2
15 a 30
Point gagné par : (match joueur) ? 2 2
15 a 30
Point gagné par : (match joueur) ? 2 2
15 a 40
```

```
Point gagné par : (match joueur) ? 2 1
30 a 40
Point gagné par : (match joueur) ? 2 1
40 a 40
Point gagné par : (match joueur) ? 2 2
Jeu gagné par Rafael
0 a 1
Point gagné par : (match joueur) ? 1 1
30 a 30
Point gagné par : (match joueur) ? 2 1
15 a 0
Point gagné par : (match joueur) ? 1 2
30 a 40
Point gagné par : (match joueur) ? 1 2
Jeu gagné par Yannick
0 a 1
Point gagné par : (match joueur) ? 2 2
15 a 15
Point gagné par : (match joueur) ? 1 1
15 a 0
...etc...
```

1. A l'origine, les points étaient comptés sur des cadrans d'horloge. Chaque balle comptait pour un quart d'heure, d'où la suite 0, 15, 30, 45. Le 45 s'est ensuite simplifié en 40.

3 ANNEXE 3 : codes

```
1 /*match.h */
2
3 #ifndef MATCH_H
4 #define MATCH_H
5
6 #define NOMBREMAXPARTIESSIMULT 4
7 #define TAILLECHAINES 30
8
9 typedef struct Jeu {
10     int balles_gagnes[2];
11 } Jeu;
12
13 typedef struct Set {
14     int jeux_gagnes[2];
15     Jeu jeu_en_cours;
16 } Set;
17
18 typedef struct Match {
19     char noms_joueurs[2][TAILLECHAINES];
20     int set_gagnes[2];
21     Set set_en_cours;
22     int termine;
23 } Match;
24
25
26 void init_jeu(Jeu *jeu);
27
28
29 void init_match(Match *match);
30
31 int incrementer_jeu(Jeu *jeu,int numJoueur,char nomJoueur[]);
32 int incrementer_set(Set *set,int numJoueur,char nomJoueur[]);
33
34 int incrementer_match(Match *match,int numJoueur);
35
36 #endif

```

```
1 /* match.c */
2 #include <stdio.h>
3 #include "match.h"
4
5 int points_d_un_jeu[4]={0,15,30,40};
6 #define NB_SET_GAGNANTS_PAR_MATCH 3
7
8 void init_jeu(Jeu *jeu)
9 {
10     jeu->balles_gagnes[0]=0;
11     jeu->balles_gagnes[1]=0;
12 }
13
14
15 void init_match(Match *match)
16 {
17     printf("nom du premier joueur ? ");
18     scanf("%s",match->noms_joueurs[0]);
19     printf("nom du second joueur ? ");
20     scanf("%s",match->noms_joueurs[1]);
21     match->set_gagnes[0]=0;
```

```

22 match->set_gagnes[1]=0;
23 match->termine=0;
24 init_set(&(match->set_en_cours));
25 }
26
27 int incrementer_jeu(Jeu *jeu,int numJoueur,char nomJoueur[])
28 {
29     jeu->balles_gagnes[numJoueur]++;
30     if (jeu->balles_gagnes[numJoueur]==4)
31     {
32         printf("Jeu gagne par %s\n",nomJoueur);
33         return 1;
34     }
35     else
36     {
37         printf("%d a %d\n",
38             points_d_un_jeu[jeu->balles_gagnes[0]],
39             points_d_un_jeu[jeu->balles_gagnes[1]]);
40         return 0;
41     }
42 }
43
44 int incrementer_set(Set *set,int numJoueur,char nomJoueur[])
45 {
46     if(!incrementer_jeu(&(set->jeu_en_cours), numJoueur,nomJoueur))
47         return 0;
48     set->jeux_gagnes[numJoueur]++;
49     init_jeu(&(set->jeu_en_cours));
50     printf("%d a %d \n",
51         set->jeux_gagnes[0],set->jeux_gagnes[1]);
52     if (set->jeux_gagnes[numJoueur]>=6 &&
53         set->jeux_gagnes[numJoueur]-set->jeux_gagnes[1-numJoueur]>=2)
54     {
55         printf("Set gagne par %s\n",nomJoueur);
56         return 1;
57     }
58     return 0;
59 }
60
61 int incrementer_match(Match *match,int numJoueur)
62 {
63     if(!incrementer_set(&(match->set_en_cours),numJoueur,
64         match->noms_joueurs[numJoueur]))
65         return 0;
66     match->set_gagnes[numJoueur]++;
67     init_set(&(match->set_en_cours));
68
69     if (match->set_gagnes[numJoueur]
70         >=NB_SET_GAGNANTS_PAR_MATCH)
71     {
72         match->termine=1;
73         printf("%s bat %s par %d sets a %d\n",
74             match->noms_joueurs[numJoueur],
75             match->noms_joueurs[1-numJoueur],
76             match->set_gagnes[numJoueur],
77             match->set_gagnes[1-numJoueur]);
78     }
79     return match->termine;
80 }

```

```

1 /* tournoi.c */
2
3 #include <stdio.h>
4 #include "tournoi.h"
5
6
7 void init_tournoi(Tournoi *tournoi)
8 {
9     int numMatch;
10    printf("Combien de matchs simultanes ? ");
11    scanf("%d",&(tournoi->nb_matches_simultanes));
12    for (numMatch=0;
13         numMatch<tournoi->nb_matches_simultanes;numMatch++)
14        init_match(&(tournoi->matches_en_cours[numMatch]));
15 }
16
17 int tournoi_est_termine(Tournoi *tournoi)
18 {
19     int termine=1;
20     int numMatch;
21     for (numMatch=0; numMatch<tournoi->nb_matches_simultanes;numMatch++)
22         termine = termine && tournoi->matches_en_cours[numMatch].termine;
23     return termine;
24 }

```

```

1 /*tennis.c */
2
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include "tournoi.h"
6
7 int main(void)
8 {
9     Tournoi le_tournoi;
10    int match,joueur;
11    init_tournoi(&le_tournoi);
12    printf("Pour indiquer un point gagne :\n");
13    printf("indiquez le numero de match (1 a %d)\n",
14           le_tournoi.nb_matches_simultanes);
15    printf("et le numero du joueur qui marque (1 a 2)\n");
16
17    do {
18        printf("Point gagne par :(match joueur) ? ");
19        scanf("%d%d",&match,&joueur);
20        incrementer_match( &(le_tournoi.matches_en_cours[match-1]),joueur-1);
21    } while (!tournoi_est_termine(&le_tournoi));
22
23    return 0;
24 }

```

FIN