

Compte Rendu :

TP Informatique : Gestion de compétition d'échecs.



Code sur : <https://github.com/A-s-a-d/tournament-management-tool>

MUHAMMAD ASAD
THOMAS MARCHAND

TD3/TP5
TD3/TP5

Sommaire

Tables des matières

Partie 1 : Le programme	3
[0] : Quitter le programme	4
[1] : Ajouter un joueur	4
[2] : Supprimer un joueur	5
[3] : Afficher l'état du tournoi	6
[5] : Saisir le vainqueur d'un match en cours.....	7
[6] : Déclarer le joueur forfait	11
[7] : Commencer le tournoi	12
Partie 2 : L'algorithme du programme C.....	13
1. Algorithme de la fonction int commencer_tournoi ()......	13
2. Algorithme de la fonction int main().	14

Partie 1 : Le programme

Toutes les variables utilisées en programme sont globales :

Variables :

Nous avons mis presque toutes les variables en globale pour renvoyer ou modifier plusieurs valeurs depuis les fonctions, l'autre façon de pour renvoyer ou modifier plusieurs valeurs depuis les fonctions est d'utiliser les pointeurs.

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <string.h>
4. #define MAX_SIZE 8
5. #define MAX_SIZE_d_f 4
6. #define MAX_SIZE_f 4
7.
8.
9. #define AC_BLACK "\x1b[30m"           //https://c-for-dummies.com/blog/?p=5270
10. #define AC_RED "\x1b[31m"            // link for colors
11. #define AC_GREEN "\x1b[32m"          //For colors of text and background
12. #define AC_YELLOW "\x1b[33m"         //For colors of text and background
13. #define AC_BLUE "\x1b[34m"           //For colors of text and background
14. #define AC_MAGENTA "\x1b[35m"        //For colors of text and background
15. #define AC_CYAN "\x1b[36m"           //For colors of text and background
16. #define AC_WHITE "\x1b[37m"          //For colors of text and background
17. #define AC_NORMAL "\x1b[m"           //For colors of text and background
18.
19. #define Fond_Noir "\x1b[40m"          //For colors of text and background
20. #define Fond_Rouge "\x1b[41m"         //For colors of text and background
21. #define Fond_Vert "\x1B[42m"          //For colors of text and background
22. #define Fond_Jaune "\x1B[43m"         //For colors of text and background
23. #define Fond_Bleue "\x1B[44m"         //For colors of text and background
24. #define Fond_Magenta "\x1B[45m"       //For colors of text and background
25. #define Fond_Cyan "\x1B[46m"          //For colors of text and background
26. #define Fond_Blanc "\x1B[47m"        //For colors of text and background
27. typedef struct joueur
28. {
29.     char nom[30];
30.     char prenom[30];
31.     unsigned date;
32.     unsigned mois;
33.     unsigned annee;
34.     int q_de_finale;
35.     int d_finale;
36.     int f_inale;
37. }Joueur;
38.
39. int nb_joueur = 5;
40. char choice;
41. char choice_t;
42. int quart_de_finale=0;
43. int match = 1;
44. int supprimer;
45. Joueur j[MAX_SIZE];
46. int i, size_of_array, position_to_delete;
47. size_of_array=MAX_SIZE;
48. int continuer = 1;
49. int c;
50. Joueur j_d_f[MAX_SIZE_d_f];
51. int demi_finale=1;
52. int match_d_f = 1;
53. int nb_joueur_d_f ;
54. Joueur j_f[MAX_SIZE_f];
55. int finale = 1;
56. int nb_joueur_f;
57. int match_f = 1;
58. int forfa;
59. int forfait;
```

[0] : Quitter le programme

Pour sortir de programme nous avons mis le programme dans un boucle while tant que l'utilisateur ne mettra pas 0 pour variable « choice » il restera en programme

```
1. while(choice!='0')
2. {
3.     /* program ici */
4. }
```

[1] : Ajouter un joueur

Pour ajouter un joueur on demande si l'utilisateur veut ajouter un joueur si oui l'utilisateur peut entrer le nom, prénom, puis la date sous format dd/mm/aaaa.

```
1. int ajouter(Joueur j[], int nb_joueur)
2. {
3.     char choix;
4.     printf("ajouter un joueur? (o pour oui/n pour non)");
5.     scanf(" %c",&choix);
6.     while(nb_joueur < 8 && choix=='o')
7.     {
8.         printf("entrer le nom pour joueur j[%d]:\n",nb_joueur);
9.         scanf(" %s", j[nb_joueur].nom); // on peut utiliser : scanf ou gets(&j[i].nom);
10.        printf("entrer le prenom pour joueur j[%d]:\n",nb_joueur);
11.        scanf(" %s", j[nb_joueur].prenom);
12.        printf("entrer la date de naissance sous format dd/mm/aaaa pour joueur j[%d]:\n",nb_joueur);
13.        scanf(" %d/%d/%d",&j[nb_joueur].date,&j[nb_joueur].mois,&j[nb_joueur].annee);
14.        nb_joueur++;
15.        printf("ajouter un joueur? (o/n)");
16.        scanf(" %c",&choix);
17.    }
18.    if(nb_joueur>=8 && choix=='o')
19.    {
20.        printf("vous avez deja ajouter 8 joueurs\n");
21.    }
22.    if(choix=='n')
23.    {
24.        printf("%svous avez choisi de ne pas ajouter un joueur %s\n",AC_CYAN,AC_NORMAL);
25.    }
26.    return nb_joueur;
27. }
```

[2] : Supprimer un joueur

Pour supprimer un joueur on affiche la liste de tous les joueurs et demande un numéro de joueur à supprimer, si le joueur existe alors il sera supprimé et les autres joueurs de tableau vont monter en tableau et nombre de joueurs sera réduit d'un, chaque fois qu'il y a un joueur supprimer.

```
1. int suppr(Joueur j[])
2. {
3.     int i, size_of_array, position_to_delete;
4.     size_of_array=nb_joueur;
5.
6.     printf("\nElements of array are \n: ");
7.     for(i=0; i<nb_joueur; i++)
8.     {
9.         printf(" %s\n",j[i].nom);
10.        printf(" %s\n",j[i].prenom);
11.        printf(" %d/%d/%d\n",j[i].date,j[i].mois,j[i].annee);
12.    }
13.
14.    printf("Enter the element position to delete : ");
15.    scanf("%d", &position_to_delete);
16.
17.    if(position_to_delete < 0 || position_to_delete > size_of_array)
18.    {
19.        printf("Invalid position! Please enter position between 1 to %d", size_of_array);
20.    }
21.    else
22.    {
23.        for(i=position_to_delete; i<size_of_array-1; i++)
24.        {
25.            j[i] = j[i + 1];
26.        }
27.        size_of_array--;
28.        nb_joueur=size_of_array;
29.    }
30.    printf("\nElements of array after delete are \n: ");
31.    for(i=0; i<size_of_array; i++)
32.    {
33.        printf(" %s\n",j[i].nom);
34.        printf(" %s\n",j[i].prenom);
35.        printf(" %d/%d/%d\n",j[i].date,j[i].mois,j[i].annee);
36.    }
37.    return nb_joueur;
38. }
```

[3] : Afficher l'état du tournoi

Pour afficher l'état de tournoi il suffit d'aller chercher la variable « finale » de structure joueur_finale, si la variable est à un pour un joueur c'est qu'il a gagné le tournoi. On peut faire la même chose avec variables « demi-finale » de structure joueur_demi_finale et « quart_de_finale » de structure pour trouver les vainqueurs des matchs pour demi-finale et quart de finale

```
1. void etat_de_tournoi()
2. {
3.
4.     for(int i=0; i == MAX_SIZE; i++)
5.     {
6.         printf("quart_de_finale : \n");
7.         printf("joueur_quart_de_finale %d a eu %d en quart de finale",i,j[i].q_de_finale);
8.     }
9.
10.    for(int i=0; i == MAX_SIZE_d_f; i++)
11.    {
12.        printf("demi_finale : \n");
13.        printf("joueur_demi_finale %d a eu %d en demi finale",i,j_d_f[i].d_finale);
14.    }
15.
16.    for(int i=0; i == MAX_SIZE_f; i++)
17.    {
18.        printf("demi_finale : \n");
19.        printf("joueur_demi_finale %d a eu %d en demi finale",i,j_f[i].f_inale );
20.    }
21. }
```

[5] : Saisir le vainqueur d'un match en cours

Pour le vainqueur on a 3 fonctions :

1^{er} pour les vainqueurs de quart de finale dans cette fonction on vérifie s'il y a plus de 4 joueurs pour être en quart de final sinon on passe à demi-finale. La fonction « quart de finale » fait les matchs en fonction de nombre de joueurs et s'il y a un joueur tout seul il gagne automatiquement. Et s'il y a l'égalité le programme reste en boucle jusqu'à qu'il n'y a pas de vainqueur.

2eme fonction pour demi-finale, dans cette fonction s'il faut avoir plus de 2 et moins de 5 joueurs. Pour cela on recopie les joueurs qui ont gagnée en quart de finale dans une autre structure demi-finale. La fonction demi-finale fait les matchs en fonction de nombre de joueurs et s'il y a un joueur tout seul il gagne automatiquement. Et s'il y a égalité le programme reste en boucle jusqu'à qu'il n'y a pas de vainqueur.

3eme fonction pour finale, dans cette fonction s'il faut avoir plus de 1 et moins de 3 joueurs. Pour cela on recopie les joueurs qui ont gagnée en demi-finale dans une autre structure finale. La fonction finale fait les matchs en fonction de nombre de joueurs et s'il y a un joueur tout seul il gagne automatiquement. Et s'il y a égalité le programme reste en boucle jusqu'à qu'il n'y a pas de vainqueur.

A la fin pour trouver le vainqueur il suffit d'aller chercher la variable « finale » de structure joueur_finale, si la variable est a un pour un joueur c'est qu'il a gagné le tournoi. On peut faire la même chose avec variables « demi-finale » de structure joueur_demi_finale et « quart_de_finale » de structure pour trouver les vainqueurs des matchs pour demi-finale et quart de finale

```
1. void dec_vinq_match_qdf()
2. {
3.     if(nb_joueur<5)
4.     {
5.         quart_de_finale = 1;
6.         demi_finale = 0;
7.     }
8.
9.     if(quart_de_finale == 0)
10.    {
11.        printf("vous etes en quart de finale\n");
12.    }
13.
14.    while (quart_de_finale == 0)
15.    {
16.        switch(match)
17.        {
18.            case 1:
19.            {
20.                printf("j[0] vs j[1]: declarer le victoire ou egalite\n");
21.                printf("j[0] gagne : mettez 1/0, j[1] gagne mettez 0/1, egalite: mettez 1/1: \n");
22.                scanf("%d/%d",&j[0].q_de_finale,&j[1].q_de_finale);
23.                if((j[0].q_de_finale == 0 && j[1].q_de_finale == 1))
24.                {
25.                    match = 2;
26.                    printf("j[1] a gagnee\n\n");
27.                }
28.                if((j[0].q_de_finale == 1 && j[1].q_de_finale == 0))
29.                {
30.                    match = 2;
31.                    printf("j[0] a gagnee\n\n");
32.                }
33.            }
34.            break;
35.
36.            case 2:
```

```

37.         {
38.             printf("j[2] vs j[3]: declarer le victoire ou egalite\n");
39.             printf("j[2] gagne : mettez 1/0, j[3] gagne mettez 0/1, egalite: mettez 1/1: ");
40.             scanf("%d / %d",&j[2].q_de_finale,&j[3].q_de_finale);
41.             printf("%d %d \n",j[2].q_de_finale,j[3].q_de_finale);
42.             if(j[2].q_de_finale == 0 && j[3].q_de_finale == 1)
43.             {
44.                 match = 3;
45.                 printf("j[3] a gagnee\n\n");
46.             }
47.             if(j[2].q_de_finale == 1 && j[3].q_de_finale == 0)
48.             {
49.                 match = 3;
50.                 printf("j[2] a gagnee\n\n");
51.             }
52.         }
53.     }
54.     break;
55.
56.     case 3:
57.     {
58.
59.         if (nb_joueur == 5)
60.         {
61.             j[4].q_de_finale=1;
62.             quart_de_finale = 1;
63.
64.             demi_finale=0;
65.             break;
66.         }
67.         if(j[4].q_de_finale != 1)
68.         {
69.             printf("j[4] vs j[5]: declarer le victoire ou egalite\n");
70.             printf("j[4] gagne : mettez 1/0, j[5] gagne mettez 0/1, egalite: mettez 1/1: ");
71.             scanf("%d / %d",&j[4].q_de_finale,&j[5].q_de_finale);
72.             break;
73.         }
74.         if(j[4].q_de_finale == 0 && j[5].q_de_finale == 1)
75.         {
76.             match = 4;
77.             printf("j[5] a gagnee\n\n");
78.             break;
79.         }
80.         if(j[4].q_de_finale == 1 && j[5].q_de_finale == 0)
81.         {
82.             match = 4;
83.             printf("j[4] a gagnee\n\n");
84.             break;
85.         }
86.     }
87. }
88.
89. break;
90.
91. case 4:
92. {
93.     if (nb_joueur == 7)
94.     {
95.         j[6].q_de_finale=1;
96.         quart_de_finale = 1;
97.         printf("j[6] a gagnee\n\n");
98.
99.         demi_finale=0;
100.        break;
101.    }
102.    if(j[6].q_de_finale !=1)
103.    {
104.        printf("j[6] vs j[7]: declarer le victoire ou egalite\n");
105.        printf("j[6] gagne : mettez 1/0, j[7] gagne mettez 0/1, egalite: mettez 1/1: ");
106.        scanf("%d/%d",&j[6].q_de_finale,&j[7].q_de_finale);
107.        break;
108.    }
109.    if(j[6].q_de_finale == 0 && j[7].q_de_finale == 1)
110.    {
111.        printf("j[7] a gagnee\n\n");
112.        quart_de_finale = 1;
113.
114.        demi_finale=0;
115.        break;
116.    }
117.    if(j[6].q_de_finale == 1 && j[7].q_de_finale == 0)
118.    {
119.        quart_de_finale = 1;
120.        printf("j[6] a gagnee\n\n");
121.
122.        demi_finale=0;
123.        break;
124.    }
125. }
126. break;
127. }
128. }

```



```

129. }
130.
131. /*-----demi finale-----*/
132. void dec_vinq_demi_final()
133. {
134.     if(nb_joueur<3)
135.     {
136.         demi_finale = 1;
137.     }
138.     if( nb_joueur_d_f>4)
139.     {
140.         demi_finale = 1;
141.     }
142.
143.     if(nb_joueur<5)
144.     {
145.         j[0].q_de_finale == 1;
146.         j[1].q_de_finale == 1;
147.         j[2].q_de_finale == 1;
148.         j[3].q_de_finale == 1;
149.     }
150.
151.
152.     for(int z=0; z == nb_joueur;z++)
153.     {
154.         if(j[z].q_de_finale == 1)
155.         {
156.             j_d_f[c] = j[z];           // memcpy(&j_d_f[c], &j[i], sizeof(Joueur));
157.             nb_joueur_d_f++;
158.             c++;
159.         }
160.     }
161.
162.
163.
164.     if(demi_finale == 0)
165.     {
166.         printf("vous etes en demi finale\n");
167.     }
168.
169.     while (demi_finale == 0)
170.     {
171.         switch(match_d_f)
172.         {
173.             case 1:
174.                 printf("j[0]_demi_f vs j[1]_demi_f: declarer le victoire ou egalite\n");
175.                 printf("j[0]_demi_f gagne : mettez 1/0, j[1]_demi_f gagne mettez 0/1, egalite: mettez 1/1: \n");
176.                 scanf("%d/%d",&j_d_f[0].d_finale,&j_d_f[1].d_finale);
177.                 if((j_d_f[0].d_finale == 0 && j_d_f[1].d_finale == 1))
178.                 {
179.                     match_d_f = 2;
180.                     printf("j[1]_demi_f a gagnee\n\n");
181.                 }
182.                 if((j_d_f[0].d_finale == 1 && j_d_f[1].d_finale == 0))
183.                 {
184.                     match_d_f = 2;
185.                     printf("j[0]_demi_f a gagnee\n\n");
186.                 }
187.                 break;
188.
189.             case 2:
190.                 {
191.                     if (nb_joueur_d_f == 3)
192.                     {
193.                         j_d_f[2].d_finale=1;
194.                         demi_finale = 1;
195.                         printf("j[2]_demi_f a gagnee\n\n");
196.                         match_d_f = 3;
197.                         finale = 0;
198.                         break;
199.                     }
200.
201.                     if(j_d_f[2].d_finale !=1 )
202.                     {
203.                         printf("j[2]_demi_f vs j[3]_demi_f: declarer le victoire ou egalite\n");
204.                         printf("j[2]_demi_f gagne : mettez 1/0, j[3]_demi_f gagne mettez 0/1, egalite: mettez 1/1: ");
205.                         scanf("%d / %d",&j_d_f[2].d_finale,&j_d_f[3].d_finale);
206.                         printf("%d %d \n",j_d_f[2].d_finale,j_d_f[3].d_finale);
207.                         break;
208.                     }
209.
210.                     if((j_d_f[2].d_finale == 0 && j_d_f[3].d_finale == 1))
211.                     {
212.                         demi_finale = 1;
213.                         printf("j[3]_demi_f a gagnee\n\n");
214.                         match_d_f = 3;
215.                         finale = 0;
216.                         break;
217.                     }
218.
219.                     if((j_d_f[2].d_finale == 1 && j_d_f[3].d_finale == 0))
220.                     {

```

```

221.         demi_finale = 1;
222.         printf("j[2]_demi_f a gagnee\n\n");
223.         match_d_f = 3;
224.         finale = 0;
225.         break;
226.     }
227. }
228.     break;
229. }
230. }
231. }
232.
233. /*-----finale-----*/
234.
235.
236. void dec_vinq_finale()
237. {
238.
239.     if(nb_joueur<3)
240.     {
241.         j[0].d_finale == 1;
242.         j[1].d_finale == 1;
243.     }
244.
245.     for(int z=0; z == nb_joueur_d_f;z++)
246.     {
247.         if(j_d_f[z].d_finale == 1)
248.         {
249.             j_f[c] = j_d_f[z];           // memcpy(&j_d_f[c], &j[i], sizeof(Joueur));
250.             nb_joueur_f++;
251.             c++;
252.         }
253.     }
254.
255.     if(nb_joueur_f<1 && nb_joueur_f>2)
256.     {
257.         finale = 1;
258.     }
259.
260.     if(demi_finale == 0)
261.     {
262.         printf("vous etes en finale \n");
263.     }
264.
265.     while (finale == 0)
266.     {
267.         switch(match_f)
268.         {
269.             case 1:
270.             {
271.                 if (nb_joueur_f == 0)
272.                 {
273.                     j_d_f[0].f_inale=1;
274.                     finale = 1;
275.                     printf("j[2]_f a gagnee\n\n");
276.                     match_f = 2;
277.                     break;
278.                 }
279.                 if(j_f[0].f_inale !=1 )
280.                 {
281.                     printf("j[0]_f vs j[1]_f: declarer le victoire ou egalite\n");
282.                     printf("j[0]_f gagne : mettez 1/0, j[1]_f gagne mettez 0/1, egalite: mettez 1/1: ");
283.                     scanf("%d / %d",&j_f[0].f_inale,&j_f[1].f_inale);
284.                     printf("%d %d \n",j_f[0].f_inale,j_f[1].f_inale);
285.                     break;
286.                 }
287.                 if((j_f[0].f_inale == 0 && j_f[1].f_inale == 1))
288.                 {
289.                     finale = 1;
290.                     printf("j[1]_f a gagnee\n\n");
291.                     match_f = 2;
292.                     break;
293.                 }
294.                 if((j_f[0].f_inale == 1 && j_f[1].f_inale == 0))
295.                 {
296.                     finale = 1;
297.                     printf("j[0]_f a gagnee\n\n");
298.                     match_f = 2;
299.                     break;
300.                 }
301.             }
302.             break;
303.         }
304.     }
305.     return;
306. }
307.

```

[6] : Déclarer le joueur forfait

Pour declarer un jouerur forfait on peut utiliser la fonction supprimer un joueur car declarer un joueur forfait reviens a même que supprimer.

```
1. int forf()
2. {
3.     int i, size_of_array, position_to_delete;
4.     size_of_array=nb_joueur;
5.
6.     printf("\nElements of array are \n: ");
7.     for(i=0; i<nb_joueur; i++)
8.     {
9.         printf(" %s\n",j[i].nom);
10.        printf(" %s\n",j[i].prenom);
11.        printf(" %d/%d/%d\n",j[i].date,j[i].mois,j[i].annee);
12.    }
13.    printf("Enter the element position to delete : ");
14.    scanf("%d", &forfa);
15.
16.    if(forfa < 0 || forfa > size_of_array)
17.    {
18.        printf("Invalid position! Please enter position between 1 to %d", size_of_array);
19.    }
20.    else
21.    {
22.        for(i=position_to_delete; i<size_of_array-1; i++)
23.        {
24.            j[i] = j[i + 1];
25.        }
26.        size_of_array--;
27.        nb_joueur=size_of_array;
28.    }
29.    printf("\nElements of array after delete are \n: ");
30.    for(i=0; i<size_of_array; i++)
31.    {
32.        printf(" %s\n",j[i].nom);
33.        printf(" %s\n",j[i].prenom);
34.        printf(" %d/%d/%d\n",j[i].date,j[i].mois,j[i].annee);
35.    }
36.    return nb_joueur;
37. }
```

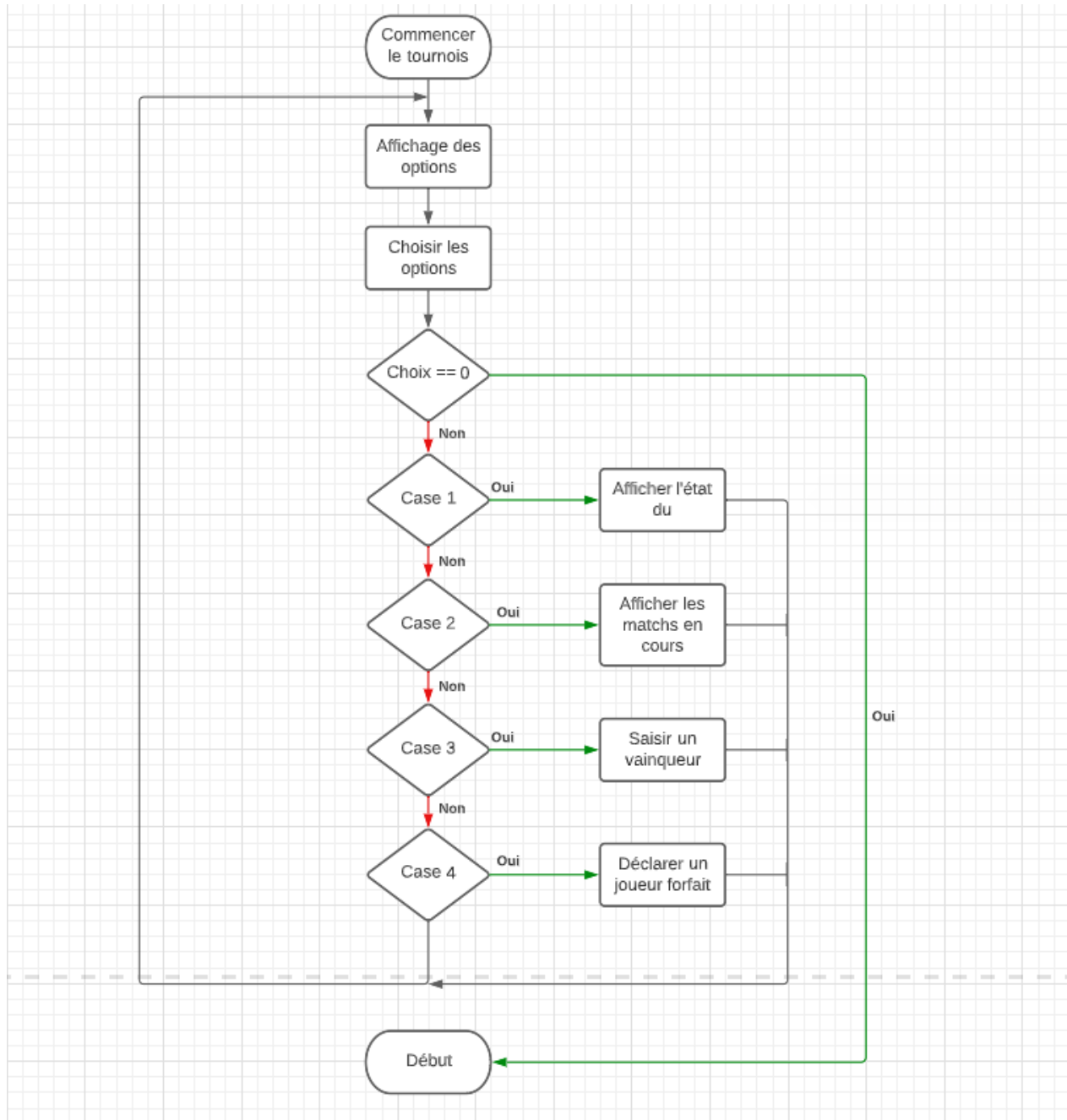
[7] : Commencer le tournoi

Pour commencer le tournoi l'utilisateur doit choisir 7, une fois commencer le tournoi l'utilisateur ne peut plus ajouter ou supprimer un joueur et avant de commencer le tournoi l'utilisateur peut qu'ajouter ou supprimer des joueurs.

```
1. int commencer_tournois(char choice, int supprimer, Joueur j[])
2. {
3.     int i, size_of_array, position_to_delete;
4.     size_of_array=MAX_SIZE;
5.
6.     do
7.     {
8.         printf("[0] Quitter le mode tournoi%s\n", AC_NORMAL);
9.         printf("[1] Afficher l'etat du tournoi\n");
10.        printf("[2] Afficher les matchs en cours\n");
11.        printf("[3] Saisir le vainqueur d'un match en cours\n");
12.        printf("[4] Declarer un joueur forfait\n");
13.
14.        printf("%sVeuillez entrer votre choix :%s\n", Fond_Blue, AC_NORMAL);
15.        scanf(" %c",&choice_t);
16.
17.        switch(choice_t)
18.        {
19.            case '1':
20.                printf("%sVous avez choisi: %s%s [1] Afficher l'etat du
tournoi%s\n", Fond_Cyan, AC_NORMAL, AC_YELLOW, AC_NORMAL);
21.                break;
22.
23.            case '2':
24.                printf("%sVous avez choisi: %s%s [2] Afficher les matchs en
cours%s\n", Fond_Cyan, AC_NORMAL, AC_YELLOW, AC_NORMAL);
25.                break;
26.
27.            case '3':
28.                printf("%sVous avez choisi: %s%s [3] Saisir le vainqueur d'un match en
cours%s\n", Fond_Cyan, AC_NORMAL, AC_YELLOW, AC_NORMAL);
29.                dec_vinq_match_qdf();
30.                dec_vinq_demi_final();
31.                dec_vinq_finale();
32.                break;
33.
34.            case '4':
35.                printf("%sVous avez choisi: %s%s [4] Declarer un joueur
forfait%s\n", Fond_Cyan, AC_NORMAL, AC_YELLOW, AC_NORMAL);
36.                forfait = forf();
37.                break;
38.        }
39.    } while(choice != '0');
40.    return(0);
41. }
42.
```

Partie 2 : L'algorithme du programme C

1. Algorithme de la fonction int commencer_tournoi ().



2. Algorithme de la fonction int main().

