



Programmation en C (EPINF2E1)

Nom et prénom :

.....

- ▶ *Durée : 2 heures.*
- ▶ *Aucun document n'est autorisé. L'usage de la calculatrice ou de tout équipement électronique est interdit.*
- ▶ *Le sujet qui vous a été attribué est unique et il ne vous sera fourni que cet exemplaire.*
- ▶ *Noircissez entièrement l'intérieur des cases choisies. Vous pouvez utiliser un correcteur liquide pour faire disparaître une case noircie par erreur (**mais ne redessinez pas la bordure de la case**).*
- ▶ *Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse. Il n'y a pas de points négatifs entre les questions.*

Codez votre numéro d'étudiant ci-dessous :

<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0
<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1
<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2
<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3
<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4
<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5
<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6
<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7
<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8
<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9

1 Questionnaire à choix multiples (12 points)

2 Questions ouvertes (8 points) – corrigé

Nous souhaitons écrire un programme qui gère le classement du championnat de France de rugby à XV, dénommé Top 14. Les équipes du Top 14 sont stockées dans un tableau de structures. On utilisera pour cela les définitions suivantes :

```
#define LGMAXNOM 26 /* longueur maximale du nom de l'équipe */
#define NBEQUIPE 14 /* nombre d'équipes du Top 14 */
struct equipe {
    char nom[LGMAXNOM+1]; /* nom de l'équipe */
    int points; /* nombre de points accumulés */
    int matchs; /* nombre de matchs joués */
};
```

Le nom de l'équipe est une chaîne de caractères. Vous pourrez vous aider des fonctions suivantes pour manipuler des chaînes de caractères :

- `int strcmp(const char *ch1, const char *ch2)` qui retourne 0 si les deux chaînes sont identiques ou un entier non nul si elles sont différentes ;
- `char *strcpy(char *dest, const char *src)` qui copie la chaîne de caractères stockée à l'emplacement désigné par `src` vers l'emplacement désigné par `dest`.

Ci-dessous, un exemple de fonction principale de test du programme :

```
int main(void)
{
    struct equipe top14[NBEQUIPE];
    char *noms_top14[NBEQUIPE] = /* tableau de constantes chaînes de caractères */
    {
        "Agen", "Bordeaux", "Brive", "Castres", "Clermont",
        "La Rochelle", "Lyon", "Montpellier", "Oyonnax", "Paris",
        "Pau", "Racing 92", "Toulon", "Toulouse" };
    initialiser_equipes(top14, noms_top14);
    resultat_match(top14, "Toulouse", 4, "Paris", 0);
    resultat_match(top14, "Montpellier", 4, "Racing 92", 1);
    resultat_match(top14, "Toulouse", 4, "Lyon", 1);
    resultat_match(top14, "Castres", 2, "Clermont", 2);
    afficher_premier_classement(top14);
    return 0;
}
```



Q.1) ♣ Écrivez la fonction d'en-tête : `void initialiser_equipes(struct equipe equipes[], char *noms_top14[])` qui initialise les structures du tableau `equipes` avec les noms contenus dans le tableau de chaînes de caractères `noms_top14`, et le nombre de points et de matchs à la valeur 0.

```
void initialiser_equipes(struct equipe equipes[], char *noms_top14[]) {  
  
    for (int i=0; i<NBEQUIPE; i++) {  
        strcpy(equipes[i].nom, noms_top14[i]);  
        equipes[i].points = 0;  
        equipes[i].matchs = 0;  
    }  
}
```

Q.2) ♣ Écrivez les deux fonctions suivantes : `int rechercher_equipe(struct equipe equipes[], char *nom)` et `void maj_equipe(struct equipe *eq, int points)`. La première fonction recherche l'équipe désignée par nom dans le tableau `equipes` et retourne l'indice correspondant ou -1 si l'équipe est absente. La seconde fonction met à jour la structure `equipe` pointée par `eq` en lui rajoutant un match et points nombre de points.

```
int rechercher_equipe(struct equipe equipes[], char *nom) {  
  
    for (int i=0; i<NBEQUIPE; i++)  
        if (strcmp(equipes[i].nom, nom) == 0)  
            return i;  
    return -1;  
}  
  
void maj_equipe(struct equipe *eq, int points) {  
  
    eq->points+=points;  
    eq->matchs++;  
}
```

Q.3) ♣ Écrivez la fonction d'en-tête : `void resultat_match(struct equipe equipes[], char *eq1, int pts1, char *eq2, int pts2)` qui met à jour, dans le tableau `equipes`, les équipes désignées par `eq1` et `eq2` en leur rajoutant respectivement `pts1` et `pts2` points et un match joué à chacune. Cette mise à jour se fera uniquement si les deux équipes existent bien dans le tableau. Pour écrire cette fonction, vous vous appuyerez sur les deux fonctions de la question précédente.

```
void resultat_match(struct equipe equipes[], char *eq1, int pts1, char *eq2, int pts2) {  
  
    int ind1 = rechercher_equipe(equipes, eq1);  
    int ind2 = rechercher_equipe(equipes, eq2);  
    if (ind1 != -1 && ind2 != -1) {  
        maj_equipe(&equipes[ind1], pts1);  
        maj_equipe(&equipes[ind2], pts2);  
    }  
}
```



Q.4) ♣ Écrivez la fonction d'en-tête : `void afficher_premier_classement(struct equipe equipes[])` qui affiche à l'écran le nom, le nombre de points et le nombre de matchs joués de la première équipe du classement, c'est-à-dire l'équipe disposant du plus de points (en cas d'égalité, pour simplifier, une seule équipe sera affichée).

```
void afficher_premier_classement(struct equipe equipes[]) {  
  
    int ind_max=0, pts_max=equipes[0].points;  
    for (int i=1; i<NBEQUIPE; i++)  
        if (equipes[i].points > pts_max) {  
            pts_max = equipes[i].points;  
            ind_max = i;  
        }  
    printf("Premiere_equipe_du_classement:\n");  
    printf("\t--> %s: %d points, %d matchs joues\n",  
           equipes[ind_max].nom, equipes[ind_max].points, equipes[ind_max].matches);  
}
```