

LIFAP1 : Session 2
Contrôle Continu Terminal (Durée : 1h30)
Mardi 18 juin 2019

***Recommandations :** Les documents, calculatrice, téléphone portable sont interdits.*

La qualité de l'écriture et de la présentation sera prise en compte dans la note finale.

*Vous veillerez à **respecter** les notations et les règles d'écriture des algorithmes vues en cours et en TD. Le barème est donné à titre indicatif.*

NOM :

.....

PRENOM :

.....

Numéro Etudiant :

.....

Partie A : Algorithmique (10 pts)

On dispose d'une phrase (de taille maximum **MAX**) dont les mots sont séparés par des ' '.

Exemple : "la_fin_de_1_annee_est_proche"

Dans tout l'exercice vous pourrez utiliser les fonctions / procédures suivantes :

- longueur (ch) qui retourne la longueur de la chaîne ch passée en paramètre,
- concatene (ch1, ch2) qui ajoute tous les caractères de ch2 à la suite de ceux de ch1,
- copie (ch1, ch2) qui copie le contenu de ch2 dans ch1.

- 1- Écrire l'algorithme d'une fonction booléenne `verifie_chaine` qui vérifie que le format de la chaîne est correct (uniquement des caractères minuscules et des ' ').

2- Écrire l'algorithme d'un sous-programme `compte_mots` permettant de compter et de retourner le nombre de mots d'une phrase passée en paramètre. Dans l'exemple le résultat serait 7.

3- On souhaite séparer et identifier les différents mots de la phrase et recopier chacun de ces mots dans un tableau de chaînes de caractères (de taille `MAX*MAX`). Écrire l'algorithme d'un sous-programme `separe_mots` qui à partir d'une chaîne de caractères passée en paramètre, remplit un tableau de chaînes de caractères avec les différents mots la constituant. Le nombre de mots dans la phrase de départ pourra être passé également en paramètre. Exemple :

la	fin	de	l	annee	est	proche			
----	-----	----	---	-------	-----	--------	--	--	--

- 4- Écrire l'algorithme d'un sous-programme `affiche_tab_mots` permettant d'afficher (en les séparant par des espaces) tous les mots contenus dans le tableau rempli à la question 3. Le nombre de mots dans la phrase de départ pourra être passé également en paramètre.

- 5- Écrire l'algorithme d'un sous-programme `reconstruit_phrase` qui, à partir du tableau de mots obtenu dans la question 3, reconstruit la chaîne de caractères de départ.

- 6- Écrire l'algorithme du programme principal qui :
- saisit une chaîne de caractères dans laquelle les mots sont séparés par des ' _ ' (on recommencera la saisie tant que le format n'est pas correct),
 - construit et affiche le tableau de mots,
 - construit et affiche la chaîne reconstituée à partir du tableau de mots.

Partie B : Langage C / C++ (10 pts)

Match Nul. On souhaite recueillir un certain nombre d'informations concernant les joueuses des équipes actuellement en lice pour la coupe du monde de football féminine.

Pour chaque joueuse, on mémorisera son nom, son prénom, son poids, et un champ booléen permettant de savoir si elle est titulaire ou non. Une équipe est connue par sa nationalité (France, Argentine, ...) ainsi qu'un tableau des 23 joueuses (11 titulaires + 12 remplaçantes).

1. Déclarer en langage C/C++ deux constantes `TAILLE_EQUIPE` et `TITULAIRE` ayant pour valeurs respectives 23 et 11.

2. Donner en langage C/C++ la définition des structures `JOUEUSE` et `EQUIPE` précédemment décrites.

```
struct joueuse                                struct equipe
{                                              {
```

3. Écrire en langage C/C++ une **fonction** `saisie_valeur` permettant de renvoyer **un réel strictement positif** choisi par l'utilisateur.

4. Écrire en langage C/C++ une **fonction** `remplir_joueuse` permettant de remplir la structure `JOUEUSE` avec son nom, son prénom et son poids (on n'indiquera pas ici si elle est ou non titulaire). On utilisera le sous-programme précédent pour s'assurer que le poids est bien une valeur strictement positive.

5. Écrire en langage C/C++ une **procédure** `ajoute_joueuse` permettant d'ajouter **une** joueuse l'`EQUIPE`, s'il reste encore de la place dans le tableau de joueuses.

6. Écrire en langage C/C++ une **procédure** `remplir_equipe` permettant de saisir la nationalité des joueuses ainsi que les informations concernant les 11 titulaires et les 12 remplaçantes de l'équipe. Attention, il faudra renseigner **ici** pour chaque joueuse si elle est titulaire ou non.

7. Écrire en langage C/C++ une procédure `affiche_titulaire` permettant d'afficher les noms des 11 titulaires d'une équipe.

8. Écrire en langage C/C++ un sous-programme `min_max_moy` permettant de calculer et de renvoyer le poids de la plus légère d'une équipe, le poids de la plus lourde, ainsi que la moyenne des poids des joueuses d'une équipe.

9. Écrire en langage C/C++ le programme principal permettant de remplir les informations concernant une équipe, d'afficher les noms des titulaires de l'équipe puis d'afficher le poids de la joueuse la plus légère, de la joueuse la plus lourde et le poids moyen des joueuses.