

LIFAP1 – Partie B – Langage C/C++

Contrôle Continu Terminal (Durée totale : 2h)

Mardi 4 janvier 2022

***Recommandations :** Les documents, calculatrice, téléphone portable sont interdits. La qualité de l'écriture et de la présentation seront prises en compte dans la note finale. Vous veillerez à **respecter** les notations et les règles d'écriture des programmes vues en cours et en TP.*

Exercice I : Mot de passe sécurisé

Pour s'assurer une meilleure fiabilité des mots de passe, les sites web imposent de plus de plus de règles à respecter dans la constitution des mots de passe. Nous allons ici vérifier quelques-unes d'entre elles : longueur, présence de caractères minuscules / majuscules / chiffres / caractères spéciaux ...

- 1- Ecrire en langage C/C++ un sous-programme `compte` qui à partir d'une chaîne de caractères `ch` passée en paramètre permet de compter et "retourner" le nombre de caractères alphabétiques majuscules, le nombre de caractères alphabétiques minuscules, le nombre de chiffres et le nombre de caractères spéciaux (tous les autres caractères par défaut) qui composent la chaîne `ch`. Vous ne devrez pas utiliser `strlen`.

NOM :

.....

PRENOM :

.....

Numéro Etudiant :

.....

- 2- Ecrire en langage C/C++ une fonction booléenne `verifie_mdp` qui à partir d'une chaîne de caractères `ch` passée en paramètre retourne vrai si `ch` a une longueur comprise entre 8 et 16 caractères et qu'elle contient au moins 1 chiffre, 1 caractère majuscule, 1 caractère minuscule et 1 caractère spécial, faux sinon. Vous ne devez pas utiliser `strlen` pour calculer la longueur de `ch`.

- 3- Ecrire en langage C/C++ le programme principal qui demande à l'utilisateur de choisir un mot de passe. La saisie sera recommencée tant que le mot de passe ne satisfait pas les contraintes précédentes.

Exercice II : Acrostiche

Un acrostiche, du grec akrostikhos (akros, haut, élevé et stikhos, le vers), est un poème, une strophe ou une série de strophes fondés sur une forme poétique consistant en ce que, lues verticalement de haut en bas, la première lettre ou, parfois, les premiers mots d'une suite de vers composent un mot ou une expression en lien avec le poème.

Un mot est caractérisé par une chaîne de caractères `m` et le nombre de syllabes `nb_syl` qui le constituent. Un vers est représenté par un tableau de mots `tab_mots` et le nombre de pieds (syllabes) `nb_pieds` des mots qui le composent.

*M eilleurs voeux à tous
E lle vous en souhaite une foulditude
I rréalisables peut être mais sincères
L a santé en tête de liste
L 'amour et le bonheur s'ensuivent
E t peut être quelques euros
U ne bonne année en résumé
R ien que pour vous les amis
S ans oublier famille et amis*

*V oir arriver deux mille dix-neuf
O uvrir sa bourse aux pauvres
E n leurs yeux voir la lumière
U ne petite flamme s'allumer
X XL sera notre satisfaction*

1- Définir 2 constantes `CHMAX` et `MAX_MOTS` ayant pour valeurs respectives 64 et 32.

2- Ecrire en langage C/C++ les structures `mot` et `vers`. On utilisera les constantes précédentes.

```
struct mot                                     struct vers
```

3- Ecrire en langage C/C++ une **procédure** `remplir_mot` qui permet de remplir et "retourner" la structure `mot` (le mot et le nombre de syllabes) en demandant les informations à l'utilisateur.

4- Ecrire en langage C/C++ une fonction `remplir_vers` qui remplira la structure `vers` avec autant de mots que nécessaire pour atteindre 12 pieds au total (un alexandrin). On devra recommencer toute la saisie si la somme des pieds des mots proposés n'est pas égale à 12. On utilisera la procédure précédente.

5- Ecrire en langage C/C++ un sous-programme `genere_acrostiche` qui à partir d'une chaîne de caractères `ch` passée en paramètre génère et "retourne" un tableau de vers, la première lettre de chaque vers étant une lettre de `ch`. On recommencera la saisie d'un vers tant que celui-ci ne commence pas par la lettre correspondante dans le mot. Dans l'exemple de la page précédente, la chaîne est : "MEILLEURS VOEUX", et le vers d'indice 3 doit commencer par la lettre "L" → "La santé en tête de liste".