

Examen théorique de programmation de septembre 2002 (1^{ère} partie).

Question 1 : (/20)

Veillez donner le code d'une fonction permettant de compter le nombre de caractères que l'on retrouve dans une chaîne dont l'adresse est passée en paramètre. Cette valeur est retournée par la fonction.

Question 2: (/20)

Veillez donner le code d'une fonction permettant de déterminer si un nombre entier passé en paramètre est un nombre pair ou impair. La fonction renvoie un booléen qui est faux si le nombre est pair et vrai si le nombre est impair.

Question 3 : (/20)

Veillez donner le code d'une fonction permettant l'insertion d'une chaîne de caractères dans une autre. La fonction reçoit comme paramètres les adresses des deux chaînes ainsi que la position à laquelle la deuxième chaîne doit être insérée dans la première. Elle ne doit rien retourner comme valeur.

Exemple:

```
char *tab1="ceci est essai";  
int pos=9;  
char *tab2="un ";  
insert(tab1,tab2,pos);  
printf("%s\n",tab1);
```

Le résultat de l'exécution de la dernière ligne de code doit donner: ceci est un essai.
Remarque: il est important que la fonction teste la validité de la position renseignée.

Question 3 : (/20)

1. A quoi servent les **prototypes** de fonctions et à quel(s) moment(s) vont ils être utiles?
2. Veuillez expliquer ce qui diffère entre les instructions **while** et **do while**.
3. Veuillez expliquer ce qui diffère entre les **variables globales et locales**.
4. A quoi peut servir le mot clef **extern** ?
5. Qu'est ce qu'un **pointeur générique**?

Examen théorique de programmation de septembre 2002 (2^{ière} partie).

Question 1 : (/20)

- Donner trois façons distinctes de pouvoir se repositionner au début d'un fichier ouvert en mode buffeurisé. Vous devez, par un exemple succinct, donner la syntaxe des instructions utilisées.
- Vous devez ouvrir un fichier au début de votre fonction main() pour y accéder en lecture/écriture. Si le fichier n'existe pas, il doit être créé tandis que s'il existe, son contenu doit être conservé. Les écritures ne sont pas nécessairement faites en fin de fichier. Veuillez donner les lignes de code permettant une telle opération.

Question 2: (/20)

Imaginons que vous devez lire les enregistrements présents dans un fichier, chaque enregistrement étant séparé du suivant par un caractère new-line. Dans un enregistrement, on retrouve des champs au format ascii séparés par une virgule (comma separator) dont voici un exemple:

Dupond,Damien,100.752
Van piperzeele,Albert,200.41

- Veuillez donner le code d'une fonction permettant de retourner le nombre d'enregistrements présents dans le fichier.
- Veuillez donner la forme de la fonction permettant la lecture d'un enregistrement dans le fichier et l'affectation des deux premiers champs dans des chaînes de caractères distinctes et le dernier champ dans un flottant.

Question 3 : (/20)

Vous devez gérer un fichier contenant des fiches sous forme de structures qui y ont été enregistrées. La structure contient les champs suivants:

```
struct membre  
{  
    char nom[25];  
    char prenom[25];  
    float dette;  
    BOOL efface;  
};
```

L'ensemble des fiches effacées ont le champ efface à la valeur vraie. C'est donc un effacement logique et non un effacement physique.

Donner le code de la fonction permettant de purger le fichier, c-à-d d'en effacer physiquement les fiches dont le champ efface est à la valeur vraie.