



# PROGRAMMATION IMPÉRATIVE

## PG-101

F. MORANDAT

Filière : Informatique

Année : 1

Semestre : 5

Date de l'examen : 21/01/2013

Durée de l'examen : 2h

**E N S E I R B**  
**M A T M E C A**  
**B O R D E A U X**

Documents : autorisés  non autorisés   
 Calculatrice : autorisée  non autorisée   
 Autre : Attention à la clarté du code et l'écriture.  
 Barème indicatif : 4 ; 2 ; 14

## 1 Fonction mystère

Que fait cette fonction ? (La réponse ne doit pas excéder une phrase)

```
void mystere(int l, int *a) {
    int i, d, t;
    do {
        d = 1;
        for(i = 0; i < l - 1 ; i++)
            if(a[i] > a[i + 1]) {
                t = a[i];
                a[i] = a [i + 1];
                a[i + 1] = t;
                d = 0;
            }
    } while(!d);
}
```

Indication essayez l'appel suivant :

```
int v[] = { 1, 3, 2 };
mystere(3, v);
```

## 2 Échange de valeurs

Nous souhaitons écrire une fonction qui, étant donnés deux pointeurs sur des chaînes de caractères, s'assure que la chaîne pointée par le premier (**f**) est toujours plus petite que celle pointée par le second (**s**) dans l'ordre lexicographique. La signature de cette fonction est imposée :

```
void order(char **f, char **s);
```

**Rappel** : pour comparer des chaînes vous pouvez utiliser la fonction `int strcmp(const char *s1, const char *s2)`.

## 3 Problème

*SVP lire jusqu'à la fin avant de commencer*

On souhaite créer une librairie de manipulation de listes de lectures (*playlists*) de fichiers musicaux.

- 1) Chaque morceau est identifié par :
  - Un nom de fichier (seule information obligatoire)
  - Un nom de titre
  - Un nom d'artiste
  - Un nom d'album
  - Un numéro de piste
  - Un genre (ne considérer que : ROCK, POP, JAZZ, HIP\_HOP)Quel est le meilleur moyen de représenter ces données ? Écrire le code correspondant et justifier vos choix.
- 2) Nous appellerons *playlist* une suite de morceaux.
  - i) En supposant que le nombre maximum de chansons est connu d'avance (`MAX_SONG`), quels sont les différents moyens de représenter une telle *playlist* ?
  - ii) Définir la constante `MAX_SONG` valant 100.
- 3) Nous supposerons qu'une *playlist* est composée d'un tableau de chansons.
  - i) Comment connaître le nombre de chansons dans la *playlist* ? Écrire la fonction `nb_song()` qui, étant donnée une *playlist*, retourne le nombre de chansons.
  - ii) Écrire la fonction `init_list()`.
  - iii) Écrire la fonction `append_song()` qui ajoute une chanson à la fin de la liste.
- 4) Écrire deux fonctions pour afficher les informations relative à une chanson.
  - i) La première ne doit afficher qu'une ligne avec les informations importantes (titre - album - artiste).
  - ii) La seconde affichera toutes les informations disponibles sur plusieurs lignes en texte lisible (c'est-à-dire sans nombre inutile). L'indication **Unknown** remplacera une information manquante dans la seconde fonction d'affichage. *Il est vivement conseillé de faire des fonctions auxiliaires.*
  - iii) Écrire la fonction `scan_song` qui permettra de saisir des informations relatives à une chanson. Vous pourrez choisir la méthode de votre choix pour indiquer une absence de valeur mais pensez à l'expliquer (au moins dans un commentaire). *Pour simplifier nous ne considérerons pas la saisie des espaces.*
- 5) À partir d'une *playlist* nous voulons pouvoir faire des sélections par style musical. Quelle structure peut représenter de telles sous-listes ? Écrire une fonction qui, étant donné une *playlist* et un genre de musique, retourne la *playlist* uniquement constituée de ce genre.
- 6) De la même manière nous aimerions pouvoir construire une liste ne contenant qu'un nom d'artiste particulier. Écrire une telle fonction.
- 7) À partir d'une *playlist*, nous aimerions pouvoir la mélanger en place (c'est à dire sans en créer une nouvelle). L'algorithme proposé est le suivant :
  - Pour chaque chanson de la liste, choisir une chanson aléatoirement n'étant pas avant et les permuter.Écrire une telle fonction.
- 8) Écrire une fonction `main()` permettant de saisir `n` morceaux (`n` sera passé en argument au programme). La suite de votre programme permettra de tester toutes ou parties de vos fonctions. **Bonus** cette question peut entrainer des points positif hors barème.