

## PROGRAMMATION C

### TP5

LICENCE MATHS-INFO  
13 FÉVRIER 2012

L'ensemble des fichiers de ce TP seront placés dans un même répertoire et avec un unique `makefile`. Même si ce n'est pas explicitement écrit, les fonctions doivent être testées dans des programmes appropriés.

Au lieu d'utiliser la fonction `printf` pour afficher à l'écran vos résultats, vous les écrirez dans un fichier texte à l'aide de `fprintf` (à ouvrir/fermer avec `fopen/fclose`) et vous joindrez ce fichier au compte-rendu de TP.

#### EXERCICE 1 : COPIE D'UNE CHAÎNE DE CARACTÈRES

Écrivez une fonction de prototype `char * strcpy(char *)` qui prend en paramètre une chaîne de caractère et retourne un pointeur sur une copie de cette chaîne.

#### EXERCICE 2 : CONVERSION EN CHAÎNE DE CARACTÈRES

Écrivez une fonction `char * int2String(int i)` qui convertit l'entier `i` en une chaîne : l'entier 100 doit être convertie en la chaîne "100".

#### EXERCICE 3 : PARAMÈTRES DE LA FONCTION MAIN

Construisez un exécutable qui prend en argument sur la ligne de commande un entier `n` et qui affiche `n` fois le contenu de la variable d'environnements `USER` si cette dernière est définie.

#### EXERCICE 4 : INTÉGRALE PAR LA MÉTHODE DES RECTANGLES

On souhaite écrire une méthode `integrale` qui prend en argument une fonction  $f$  de  $\mathbb{R}^R$ , un intervalle  $[a, b]$  et un entier  $n$  puis calcule l'approximation de l'intégrale  $\int_a^b f(x)dx$  par la méthode des rectangles (en utilisant  $n$  rectangles).

Quelle est l'en-tête de la fonction `integrale` ?

Codez cette fonction, puis testez-là (par exemple en comparant le résultat du calcul de  $\int_a^b \cos$  avec  $\sin(b) - \sin(a)$ ).

Rappel sur la méthode des rectangles :

$$\int_a^b f(x)dx \approx \sum_{n=0}^{N-1} (x_{n+1} - x_n) f\left(\frac{x_n + x_{n+1}}{2}\right)$$

où  $x_n = a + n \times \frac{b-a}{N}$ .

## EXERCICE 5 : TABLEAU DE FONCTIONS

Ecrivez un programme `calfon` permettant d'obtenir les valeurs des principales fonctions transcendantes sur les réels : `sin`, `cos`, `exp`, `log`, etc. (voir `math.h`). Vous utiliserez un tableau de fonctions, ainsi que les arguments de la fonction `main`. La comparaison de chaînes de caractères pourra se faire en utilisant les fonctions de la bibliothèque standard déclarées dans `string.h`.

Exemple d'utilisation (> est l'invite du système) :

```
> calfon sin 0.5
0.479426
> calfon sqrt 4
2.000000
> calfon log10 23
log10: fonction inconnue
>
```

## EXERCICE 6 : TRI DES LIGNES D'UN TEXTE

Ecrivez une fonction `char * ranger(char * texte)` qui reçoit pour argument une chaîne de caractères représentant un texte formé de plusieurs lignes. Chaque ligne (y compris la dernière) se termine par un caractère `\n` qui (sauf pour la dernière) la sépare de la ligne suivante :

```
"Damien\nBeatrice\nEmile\nGael\nJean\nIlona\nAnne\nHenri\nFrancis\nCarole\n"
```

Cette fonction renvoie une chaîne de caractères nouvellement allouée, contenant le texte donné après que les lignes ont été rangées par ordre croissant. Pour l'exemple ci-dessus, cela donne :

```
"Anne\nBeatrice\nCarole\nDamien\nEmile\nFrancis\nGael\nHenri\nIlona\nJean\n"
```

Pour faire le tri, on utilisera la fonction `qsort` de la bibliothèque standard (cf. `stdlib.h`), dont l'en-tête est :

```
void qsort(void *tab, int nbr, int taille, int (*comp)(const void *, const void *));
```