

TD MÉTHODOLOGIE DE LA PROGRAMMATION FEUILLE D'EXERCICES N°2

Sebastien.Varrette@imag.fr (Bureau BR.4.07)

1 Préliminaires

Vous êtes tenus cette fois-ci de soumettre vos travaux. Pour cela, on utilisera la plateforme Moodle du SMA disponible à l'adresse <http://claudes-berge.central.lu/moodle/>

Vous avez jusqu'à **lundi 24 octobre, 8h00** pour uploader l'archive de vos sources au format **tgz**. Rappel des différentes commandes :

- Pour créer le répertoire qui contiendra vos sources : `mkdir TD2`
- une fois votre travail achevé, créer l'archive tgz de votre répertoire de la façon suivante : `tar cvzf TD2_nom_prenom.tgz TD2/`

ATTENTION! Veillez à supprimer les fichiers de backup (*~) avant de créer l'archive. Tout autre format de compression (zip, rar etc...) ne sera pas corrigé.

2 Opérateurs en C

Ecrivez un programme `operateurs_int.c` demandant deux entiers x et y et affiche le résultat des opérations suivantes :

```
x + y   x - y   x * y   x / y
x <= y  x >= y  x == y  x != y
x && y  x || y   !x
x & y   x | y   x ^ y   ~x
x << 2  x >> 2
```

3 Structures de contrôle et boucles

1. Ecrire le programme `time_management.c` comportant :
 - la déclaration de 3 variables globales entières `heure`, `minute` et `seconde`;
 - la procédure `printTime` qui affiche le message suivant :
Il est ... heure(s) ... minute(s) ... seconde(s)
en respectant l'orthographe du singulier et du pluriel;
 - la procédure `setTime` prenant en paramètre les entiers `h`, `m` et `s` et affecte les valeurs des variables globales `textttheure`, `minute` et `seconde`.
 - la procédure `tick` qui incrémente l'heure de 1 seconde
 - la fonction `main` qui illustre l'utilisation des procédures précédentes.
2. Écrire le programme `allnumber.c` définissant les procédures suivantes :
 - `void printInc(unsigned int n)` qui affiche tous les entiers de 0 à `n` en utilisant une boucle `for`.

- void `printInc2(unsigned int n)` qui affiche tous les entiers de 0 à n en utilisant une boucle `while`.
- void `printDec(unsigned int n)` qui affiche tous les entiers de n à 0 en utilisant une boucle `for`.
- void `printDec2(unsigned int n)` qui affiche tous les entiers de n à 0 en utilisant une boucle `while`.

La fonction `main` devra illustrer l'utilisation de ces fonctions après avoir demandé à l'utilisateur de saisir un entier.

3. Écrire un programme `triangle.c` qui affiche un triangle rempli d'étoiles, s'étendant sur un nombre de lignes entré au clavier. Exemple :

```

Nombre de lignes:  5
*
**
***
****
*****

```

4. Écrire un programme `moyenne_note.c` qui permet de lire une série de notes comprises entre 0 et 20 et qui affiche ensuite la moyenne de ces notes. La fin de la saisie se fera sur la note égale à 99. Exemple :

```

note 1 : 5
note 2 : 14
note 3 : 12
note 4 : 7
note 5 : 99
--> Moyenne des 4 notes entrées : 9.5

```

5. Écrire un programme `somme.c` qui demande un entier n et affiche la somme

$$S_n = 1 + 2 + \dots + n = \sum_{k=1}^n k$$

6. Écrire un programme `somme_ak.c` qui demande un entier n et un flottant a et affiche la somme

$$S_n(a) = 1 + a + \dots + a^k = \sum_{k=1}^n a^k$$

7. Écrire un programme `reverse.c` qui demande un entier n et affiche n en inversant l'ordre des chiffres.