

TP n° 02 - Module 1207

Ce TP va porter sur les boucles et les conditions, puis introduire les fonctions.

1. Finir le TP n° 01.

2. Boucles et affichages textes :

- a) Écrire le programme `printf1.c` qui affiche ceci :

```
*
**
***
****
```

- b) Écrire le programme `printf3.c` qui affiche ceci :

```
*
***
*****
*****
```

- c) Écrire le programme `printf6.c` qui affiche les valeurs d'un tableau d'entier (`int tab[6]={5,9,7,3,14,8};`) en horizontal. La première valeur du tableau est `tab[0]`, la seconde `tab[1]`, etc. On affiche ligne par ligne les valeurs avec autant * que la valeur :

```
*****
*****
*****
***
*****
*****
```

3. Jeu de recherche d'un nombre :

- a) Écrire un programme `cherche1.c` qui permet de deviner un nombre entier, inconnu du joueur, compris entre 0 et 100 inclus. Ce nombre sera défini dans le source du programme ou entré en utilisant le clavier.
- b) La librairie `stdlib.h` contient, entre autres, la fonction `int rand()` qui génère un nombre entier aléatoire entre 0 et `RAND_MAX`. Utiliser cette fonction pour générer le nombre à découvrir (créer le fichier `cherche2.c`).

4. Conversion °F en °C :

- a) Écrire un programme (`celsius1.c`) qui permette de convertir une température de °F en °C sachant que la relation mathématique entre les 2 unités est linéaire (du type $y=ax+b$) et que l'on a les propriétés suivantes : 32 °F correspond à 0 °C et 212 °F correspond à 100 °C. Définir (avant le `main()`) et utiliser une fonction pour réaliser cela.

5. Fonctions pour tableaux et chaînes :

- a) Écrire sous forme de fonctions (dans le fichier `tableaux.c`), les exercices du TD n°02.

```
void afficheTableauEntier(int *tab, int len);
int chercheMaximum(int *tab, int len);
int cherchePositionMinimum(int *tab, int len);
void permuteTableauEntier(int *tab, int len, int i1, int i2);
int longueur(char *str);
void recopie(char *src, char *dst);
void recopieALenvers(char *src, char *dst);
void concatene(char *src1, char *src2, char *dst);
```

- b) Tester toutes ces fonctions dans un `main()`.