Travaux Pratiques de langage C n°1

Cours d'Informatique de l'École Centrale d'Électronique

—Ingénieur Première Année—

Le but de ce TP est de vérifier que les concepts vus en cours ont bien été assimilés :

- syntaxe du langage *C* ;
- introduction aux types de base ;
- entrées/sorties ;
- structures de boucles et structures conditionnelles.

▸ **Exercice 1.** *Commenter le programme suivant :*

```
#include <stdio.h>

void main(void)
{ printf("Bonjour\n");
}
```

▸ **Exercice 2.** *Écrire un programme qui déclare une variable* `entier` *de type* `int`*, lui affecte la valeur 3 et l'affiche. Refaire la même chose pour une variable de type* `float`*.*

▸ **Exercice 3.** *Nous nous intéressons à la fonction* `scanf` *de la bibliothèque d'entrées/sorties standards. Cette fonction effectue la lecture des caractères se trouvant sur l'entrée standard et les interprète selon un format précis. Commenter le programme suivant :*

```
#include <stdio.h>

void main(void)
{ int n;

  printf("Entier : ");
  scanf("%d", &n);
  printf("n=%d\n", n);
}
```

Écrire un programme qui lit au clavier un `float`*, un* `char`*, un* `int` *et qui les affiche à l'écran.*

▸ **Exercice 4.** *Écrire un programme qui lit un caractère entré au clavier, qui l'affiche ainsi que son code ASCII.*

▸ **Exercice 5.** *Dans cet exercice, nous allons utiliser la structure conditionnelle. Écrire un programme qui lit un entier entré au clavier et qui affiche si ce nombre est pair ou impair.*

▸ **Exercice 6.** *On se propose de résoudre cet exercice en utilisant des structures de boucle. Écrire un programme qui calcule les carrés des* `n` *premiers entiers (on vérifiera la correction du programme pour* `n=0`*) : en utilisant une boucle* `for` *puis en utilisant une boucle* `while` *et enfin en utilisant une boucle* `do-while`*.*

▸ **Exercice 7. (table ASCII)** *Donner une fonction de nom* `TableAscii` *affichant la table des codes ASCII complète (c.-à-d. les 256 codes possibles). On utilisera le format d'affichage (plus économique) suivant :*

Car	code	Car	code	Car	code	Car	code
⋮							⋮
'	39	g	103	§	167	ç	231
(	40	h	104	¨	168	è	232
)	41	i	105	©	169	é	233
⋮							⋮

▸ **Exercice 8. (table de multiplication)** *L'objectif de cet exercice est de générer automatiquement les tables de multiplications sur les entiers de* `1` *à* `k` *:*

1. *donner une fonction* `AffMult` *prenant deux entiers* `n` *et* `m` *en arguments. La fonction affichera le résultat de la multiplication de* `n` *par* `m` *suivant le format :* `3 * 4 = 12`*;*
2. *on souhaite maintenant afficher une ligne complète de la table de multiplication. Donner une fonction* `LigneMult` *prenant en arguments deux entiers* `n` *et* `k`*. La fonction devra calculer et afficher sur une seule ligne les résultats de* $\mathtt{n}\times i, (i = 1, \ldots, \mathtt{k})$*.*
3. *utiliser cette fonction pour écrire la fonction* `TablesMult` *à un paramètre* `k`*. Cette fonction affichera les produits* $i \times j$*, pour* i *et* j *variant de 1 à* `k`*. Par exemple,* `TablesMult(4)` *produira l'affichage suivant :*

	1	2	3	4
1	1	2	3	4
2	2	4	6	8
3	3	6	9	12
4	4	8	12	16

▸ **Exercice 9. (recherche des nombres premiers)** *On se propose de calculer simplement les premiers nombres premiers. On rappelle qu'un nombre est dit "premier" si ses seuls diviseurs sont 1 et lui-même. À noter que par convention, 1 n'est pas considéré comme premier.*

1. *écrire une fonction* `EstPremier` *qui teste si un entier est premier ;*
2. *en déduire une fonction de nom* `PremiersPremiers` *prenant en argument un entier* `k`*, calculant et affichant les* `k` *premiers nombres premiers. Par exemple, l'appel* `PremiersPremiers(7)` *produira l'affichage :* `2, 3, 5, 7, 11, 13, 17`.