Travaux Dirigés d'Algorithmique n°1

Cours d'Informatique de Deuxième Année

—Deug SSM M3-MI—

Ce premier TD constitue une introduction au langage *C*.
L'objectif est de comprendre les concepts-clefs suivants :

- syntaxe du langage *C* ;
- le langage *C* par rapport au langage *Pascal* ;
- introduction aux types de base ;
- entrées/sorties ;
- structures de boucles et structures conditionnelles ;

▸ **Exercice 1.** *Que fait le programme suivant :*

```
#include <stdio.h>
int biz(int u, int v)
{ int t;
  while (u>0)
  { if (u<v)
    { t=u;
      u=v;
      v=t;
    }
    u=u-v;
  }
  return v;
}

void main(void)
{ int x=5, y;
  y=21;
  printf("%d %d %d\n", x, y, biz(x,y));
}
```

▸ **Exercice 2.** *Ecrire un programme demandant à l'utilisateur de rentrer un caractère. Le programme stockera la valeur de ce caractère dans une variable* `c` *de type* `char`*, la recopiera dans une variable* `i` *de type* `int`*. Enfin, le programme affichera la valeur du caractère et son code ASCII. On utilisera le format d'affichage :* `Caractere:  x Code ASCII: xxx`

▸ **Exercice 3. (structure conditionnelle)** *Ecrire une fonction* `CAPS` *lisant un caractère au clavier. La fonction devra afficher le caractère et son code ASCII seulement si c'est une minuscule*[1]*. Si c'est une majuscule, on la transformera en minuscule puis on fera le même travail d'affichage. Enfin, si c'est un autre type de caractère, on affichera juste son code ASCII.*

▸ **Exercice 4.** *Ecrire un programme permettant de lire des caractères au clavier. La saisie s'arrêtera dès que l'utilisateur tapera le caractère de fin de ligne (*`'\n'`*). Le programme devra alors afficher le nombre total de caractères lus, le nombre de chiffres, le nombre de minuscules et le nombre de majuscules.*

▸ **Exercice 5.** *Donner un algorithme de passage d'une écriture en base dix d'un entier vers une écriture en base deux. En déduire une fonction de nom* `DixVersDeux` *prenant en argument un entier* `val`*. La fonction devra afficher la décomposition en base deux de la valeur décimale de* `val`*.*

▸ **Exercice 6. (conversion base deux vers base dix)** *On cherche dans cet exercice à réaliser l'opération inverse.*

(a) *donner l'algorithme de conversion d'un nombre binaire en un nombre décimal ;*

(b) *en déduire une fonction C de nom* `DeuxVersDix` *réalisant cette opération. La fonction devra lire une suite de caractères* `0` *ou* `1` *au clavier jusqu'à ce qu'elle rencontre un* `'\n'`*. La fonction calculera au fur et à mesure la valeur décimale correspondant au nombre binaire représenté par les premiers chiffres rencontrés. La fonction retournera la valeur décimale ainsi calculée ;*

(c) *finalement, donner le programme complet appelant la fonction* `DeuxVersDix` *et réalisant l'affichage de la valeur obtenue.*

▸ **Exercice 7.** *Exprimer avec un autre type de boucle, les instructions suivantes : on précisera par ailleurs la valeur de la variable* `i`[2] *à la sortie de chaque boucle.*

(a)
```
for(i=5;i>0;i=i-1) {...}
```

(b)
```
i=10;
while(i>0 || i%2) i=i-1;
```

(c)
```
i=10;
while(i>0 && !(i%2)) i=i-1;
```

(d)
```
i=0;
do { i=i+1; ...} while(i!=5);
```

[1] Les minuscules sont stockées dans le code *ASCII* de manière contigüe et dans l'ordre alphabétique.

[2] On supposera que `i` est une variable entière signée.