

## Informatique S1 – Programmation C

### Exercices – Corrigés

#### **TD 8 : codage binaire et caractères ASCII**

Dans ce TD, nous allons réaliser des exercices couvrant la représentation des caractères en langage C.

Note :

- Avant de réaliser les exercices ci-dessous, changer dans le terminal le codage des caractères vers « Occidental » (menu Terminal → Définir le codage des caractères → Occidental (ISO-8859-15)).
- Aucun exercice sur la conversion binaire ↔ décimal n'est proposé dans la TD. Si vous pensez qu'il est important d'ajouter ce type d'exercice, n'hésitez pas à le faire.

#### **Exercice 1**

Pas de difficulté particulière, il faut juste convertir le caractère lu en numéro entier.

```
#include <stdio.h>

/* TD 8 : char-ascii*/

int main () {
    char c;
    int ascii;

    printf ("Entrez un caractere : ");
    c = getchar();
    ascii = (int) c;

    printf ("Code ASCII de %c est %d\n",c,ascii);
}
```

#### **Exercice 2**

a) Solution la plus simple : une boucle for avec un printf.

```
#include <stdio.h>

/* TD 8 : afficher ASCII*/

int main () {
    int i;
    int j;

    for (i=0; i<256; i++) {
        printf ("%d = %c \n", i,i);
    }
}
```

- b) Exercice qui rappelle certains des concepts vu lors de la TD 5. Pour afficher dans un tableau de 10 colonnes, il faut afficher le caractère '|' après chaque caractère ASCII afin de séparer les colonnes, et afin de séparer les lignes du tableau, il faut compter les colonnes (si colonne divisible par 10, on imprimer le caractère de nouvelle ligne '\n').

*Note* : certain caractères, surtout les caractères de contrôle, ne vont pas s'afficher correctement, et les premières lignes du tableau ne seront pas forcément alignées aux lignes postérieures (la séparation des colonnes avec le '|' ne sera pas forcément alignée).

```
#include <stdio.h>

/* TD 8 : char-ascii */

int main () {
    int i;

    /* afficher l'ensemble des caracteres ASCII
     * dans un tableau de 10 colonnes */
    putchar ('|');
    for (i=0; i<256; i++) {
        printf (" %c |", i);
        if (i%10 == 0) {
            putchar ('\n');
            putchar ('|');
        }
    }
}
```

### **Exercice 3**

- a) Exercice dérivé de l'exercice 8 de la TD 5 (afficher un arbre de x lignes de hauteur). La seule différence réside sur le choix du caractère, le caractère 174 (un « sur windows, ou un ® sous linux). Attention à bien changer le codage comme mentionner o début de la TD.

```
#include <stdio.h>

/* TD 8 : Arbre de Noel */

int main () {
    int i, j;
    int hauteur=10;
    char c = (char) 174;

    /* exercice c: un triangle centralise */
    for (i=1; i<=hauteur; i++)
    {
        for (j=1; j<=hauteur-i; j++) {
            putchar(' ');
        }
        for (j=1; j<=2*i-1; j++) {
            putchar(c);
        }
        putchar('\n');
    }
}
```

- b) Similaire à l'exercice précédent, mais il faut ajouter un « pied » de 4 lignes dans l'arbre. Pour cela, une boucle supplémentaire est nécessaire, une fois que le reste de l'arbre a été affiché.

```
#include <stdio.h>

/* TD 8 : Arbre de Noel */

int main () {
    int i, j;
    int hauteur=10;
    int pied=4;
    char c = (char) 174;

    /* exercice c: un triangle centralise */
    for (i=1; i<=hauteur; i++)
    {
        for (j=1; j<=hauteur-i; j++) {
            putchar(' ');
        }
        for (j=1; j<=2*i-1; j++) {
            putchar(c);
        }
        putchar('\n');
    }

    for (i=1; i<=pied; i++) {
        for (j=1; j<=(hauteur+pied)/2; j++) {
            putchar (' ');
        }

        for (j=1; j<=pied; j++) {
            putchar(c);
        }
        putchar ('\n');
    }
}
```

#### **Exercice 4**

- a) Cet exercice introduit une complexité supplémentaire : les étudiants, à chaque lecture du clavier, doivent lire 2 caractères, celui fourni par l'utilisateur et le caractère de nouvelle ligne qui reste enregistré dans le buffer de lecture.

```
#include <stdio.h>

/* TD 8 : char-ascii */

int main () {
    char c;
    char op, nl;

    op = 'O';

    do {
        printf ("Entrez un caractere : ");
        scanf ("%c%c", &c, &nl); //on capture le caractere et le nl
        printf ("Code ASCII de %c est %d\n",c,c);

        printf ("Continuer [ O | N ] ? ");
    }
```

```
scanf ("%c%c", &op, &nl); //on capture le caractere et le nl  
} while (op=='o' || op=='O');  
}
```

- b) A partir de l'exercice précédent, ils doivent identifier et compter les majuscules, minuscules et chiffres. Pour cela, ils peuvent soit utiliser soit l'exercice de *switch* fait dans la TD précédente, soit utiliser les caractéristiques propres au code ASCII (les majuscules sont codées entre 65 et 90, les minuscules entre 97 et 122, les chiffres entre 48 et 57).

```
#include <stdio.h>  
  
/* TD 8 : char-ascii */  
  
int main () {  
    char c;  
    char op, nl;  
    int maj, min, chiffre;  
    int ascii;  
  
    op = 'O';  
    maj = 0;  
    min = 0;  
    chiffre = 0;  
  
    do {  
        printf ("Entrez un caractere : ");  
        scanf ("%c%c", &c, &nl); //on capture le caractere et le nl  
        ascii = (int) c;  
  
        printf ("Code ASCII de %c est %d\n",c,ascii);  
  
        //tous les chiffres sont codes entre 48 et 57 en ASCII  
        //tous les maj sont codes entre 65 et 90  
        //tous les min sont codes entre 97 et 122  
        if (ascii>=48 && ascii<=57) chiffre++;  
        else if (ascii>=65 && ascii<=90) maj++;  
        else if (ascii>=97 && ascii<=122) min++;  
  
        printf ("Continuer [ O | N ] ? ");  
        scanf ("%c%c", &op, &nl); //on capture le caractere et le nl  
    } while (op=='o' || op=='O');  
  
    printf ("Vous avez entrer : %d chiffres, \n\t %d lettres majuscules"  
           " \n\t %d lettres minuscules\n", chiffre, maj,  
    min);  
}
```