Université d'Aix-Marseille III - Faculté des Sciences et Techniques de Saint-Jérôme

Licence Sciences et Technologies - Mention Maths-Info

I2 - Architecture des machines et des systèmes

TP n°1

Exercice 1 : Entiers décimaux, hexadécimaux et code ASCII

Ecrivez et exécutez le programme suivant. Interprétez le résultat.

```
#include<stdio.h>
#include<stdlib.h>

main()
{
    int i;
    for(i = 0; i < 128 ; i=i+1)
      printf("<%d  %x %c>\n", i, i, i);
}
```

Voici le code ASCII de quelques caractères de contrôle ainsi que leur effet : 8 → effacement du caractère précédent et recul [backspace], 9 → tabulation, 10 → passage à la ligne suivante [line feed], 11 → tabulation verticale, 12 → page suivante [form feed], 13 → retour chariot : on revient en début de ligne, 27 → escape : produit un effet en conjonction avec le caractère suivant (ici, rien), 32 → espace, 127 → delete.

Exercice 2 : Formats d'affichage

Ecrivez et exécutez le programme suivant. Interprétez le résultat.

```
#include<stdio.h>
#include<stdlib.h>

main()
{
    int n = 100;
    int z = -1;
    char c = 'a';
    float f = 0.12;
    float g = 0.0000012;

    printf("%d %u %x %c %f %e\n", n, n, n, n, n, n);
    printf("%d %u %x %c %f %e\n", z, z, z, z, z, z);
    printf("%d %u %x %c %f %e\n", c, c, c, c, c, c);
    printf("%d %u %x %c %f %e\n", f, f, f, f, f, f);
    printf("%d %u %x %c %f %e\n", g, g, g, g, g, g);
}
```

Voici le sens des formats d'affichage utilisés ci-dessous, valables pour la fonction `printf` : %d → entier relatif, %u → entier naturel, %x → hexadecimal, %c → caractère, %f → flottant avec virgule, %e → flottant avec virgule et exposant.

Exercice 3 : Conversions entre nombre décimaux, binaires et hexadécimaux
Vous allez écrire quatre fonctions C effectuant les types de conversion suivants : du binaire au décimal, de l'hexadécimal au décimal, ainsi que les deux conversions inverses. On ne considérera ici que des entiers naturels.
On mémorisera les entiers sous forme décimal dans des variable de type `int` et on mémorisera les entiers binaires et hexadécimaux dans des chaînes de caractères.
On définit donc les types suivants :

```
typedef char binaire[33];
typedef char hexa[9];
```

Ecrivez ces quatre fonctions de conversions, dont les en-têtes sont `int bin2dec(binaire b)`, `int hex2dec(hexa h)`, `void dec2bin(int n, binaire b)` et `void dec2hex(int n, hexa h)`. Puis, utilisez ces fonctions dans un programme qui demande d'entrer un format de départ (parmi le décimal, le binaire et l'hexadécimal), puis demande d'entrer un entier, puis demande le format de conversion, puis affiche l'entier précédemment entré dans ce nouveau format.

Exercice 4 : Représentation des entiers relatifs
Dans cet exercice, on souhaite représenter des entiers relatifs en utilisant une représentation par complément à 2. En utilisant les mêmes types qu'à l'exercice 3, réécrire les deux fonctions C `dec2bin` et `bin2dec` qui permettent de passer de la base 10 à la base 2 et réciproquement. Les en-têtes de ces fonctions sont les mêmes que dans l'exercice 3.

Exercice 5 : Opérations arithmétique en base 2
On souhaite pouvoir réaliser certaines opérations arithmétiques sur les entiers relatifs qui seront représentés en base 2 par un codage par complément à 2. Pour cela, il est formellement interdit de changer de base (hormis pour la saisie et l'affichage des entiers qui pourront être effectués en base 10).

(i) Ecrire la procédure `void somme (binaire a,binaire b,binaire s)` qui prend en entrée deux nombres binaires `a` et `b` et qui calcule la somme `s` de `a` et de `b`.

(ii) Ecrire la procédure `void oppose (binaire a,binaire o)` qui prend en entrée un nombre binaire `a` et qui calcule l'opposé `o` de `a`.

(iii) Ecrire la procédure `void soustraction (binaire a,binaire b,binaire d)` qui prend en entrée deux nombres binaires `a` et `b` et qui calcule le résultat `d` de la soustraction de `a` par `b`.

(iv) Ecrire la procédure `void mult (binaire a,binaire b,binaire p)` qui prend en entrée deux nombres binaires `a` et `b` et qui calcule le produit `p` de `a` et de `b`.

(v) Ecrire la procédure `void division (binaire a,binaire b,binaire q,binaire r)` qui prend en entrée deux nombres binaires `a` et `b` et qui calcule le quotient `q` et le reste `r` de la division euclidienne de `a` par `b`.