

Exercice 1:

Qu'afficherait le programme suivant

```
1  #include <stdio.h>
2  int main (void)
3  {
4      unsigned short entier;
5      unsigned char c;
6      printf ("\n Merci de rentrer un entier positif et ensuite de rentrer un caractère : ");
7      scanf ("%hu", &entier);
8      c=getchar();
9      printf ("\n L'entier rentré est : %d ", entier);
10     printf ("\n Le caractère rentré est : %c ", c);
11     return (0);
12 }
```

si l'utilisateur saisit au clavier :

- 23C (un entier 23 suivi immédiatement de la lettre C)
- 23 C (un entier 23, puis un espace, le tout suivi de la lettre C)
- 23\nC (un entier 23, puis un retour à la ligne et enfin la lettre C)

Exercice 2:

Donner le résultat de l'affichage du programme suivant, en complétant d'abord les formats d'affichage (...) de manière appropriée :

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

```
int main(void)
{
    char a = 'A';
    int b = -4, c = 2;
    unsigned int d = 2;
    double x = 1.75;

    /* Une fois affiche comme entier */
    printf("a + 2 (en entier) = ...\n", a + 2);

    /* Une fois affiche comme caractere */
    printf("a + 2 (en caractere) = ...\n", a + 2);

    printf("a = ...\n", a);
    printf("c + 3l = ...\n", c + 3l);
    printf("c + 013 = ...\n", c + 013);
    printf("c + 0x1F = ...\n", c + 0x1F);
    printf("c + x = ...\n", c + x);

    /* Les comparaisons */
    printf("b > c = ...\n", b > c);
    printf("b > d = ...\n", b > d);

    return 0;
}
```

Commentez en particulier le résultat des deux dernières expressions.

Exercice 3:

- Ecrire un programme C qui lit deux entiers positifs A et B et les échange. Il est possible d'utiliser une variable intermédiaire.
- Reprendre l'exercice précédent mais sans utiliser de variables intermédiaires.

Exercice 4:

En utilisant les opérations bits à bit, écrire un programme C qui :

- lit un entier positif et
- affiche le troisième bit, de poids le plus faible, de cet entier.

Par exemple, si le nombre lu est 12 le programme affichera 1.

Exercice 5:

Soit i une variable de type "unsigned int".

- Compléter le tableau suivant avec $i=14$:

Opérations bit à bit	valeur de l'expression en binaire	valeur en décimal
i		
$i << 1$		
$i << 4$		
$i >> 1$		
$i >> 4$		
$i \& 1$		
$i \& 4$		
$1 << 1$		
$1 << 4$		

Nous supposons que la valeur de i n'est pas modifiée d'une étape à une autre.

- Ecrire un programme C qui confirme vos réponses.

Exercice 6:

Ecrire un programme C qui lit deux entiers positifs A et B et les échange (sans variables intermédiaires) et en utilisant des opérateurs binaires.

Indications :

$$A \oplus A = 0$$

et

$$A \oplus 0 = A.$$

(en C, le xor est représenté par le symbole ^).

Exercice 7:

Que fait le programme suivant :

```
#include <stdio.h>
int main (void)
{
    unsigned char a;
    printf ("\n Merci de saisir un petit entier positif");
    printf (" compris entre 0 et 255 : ");
    scanf ("%hhu",&a);
    a=(a&0x55)+((a>>1)&0x55);
    a=(a&0x33)+((a>>2)&0x33);
    a=(a&0x0F)+((a>>4)&0x0F);
    printf ("xxxxxx %d \n", a);
    return(0);
}
```