

## TD 2 de langage C

### Le compilateur : gcc

La commande pour lancer le compilateur C est  
`gcc <fichier-source.c> -o <fichier-exécutable>`

Si on ne précise pas le nom du fichier exécutable que l'on veut obtenir, le système l'appelle par défaut `a.out`.

Pour plus de renseignements, tapez `man gcc`.

### SYNOPSIS

```
gcc [-c|-S|-E] [-std=standard]
    [-g] [-pg] [-Olevel]
    [-Wwarn...] [-pedantic]
    [-Idir...] [-Ldir...]
    [-Dmacro[=defn]...] [-Umacro]
    [-foption...] [-mmachine-option...]
    [-o outfile] infile...
```

### types, opérateurs de base, affichage, saisie

#### Exercice 1 :

- vous devez mémoriser 10000 valeurs numériques. Quelle est la quantité d'espace mémoire à prévoir ?
- Quelle est la quantité d'espace à prévoir si vous savez que ces valeurs sont des entiers compris entre -20000 et 10000 ?

#### Exercice 2 :

Déterminez le nombre minimum d'octets nécessaires pour mémoriser les variables suivantes :

```
char reponse, car ;
unsigned int nombre ;
double somme, moyenne ;
int nombreTotal ;
short int compteur ;
```

#### Exercice 3 :

Calculez à la main la valeur de chacune des variables après chaque instruction pour les 2 programmes suivants.

Puis vérifiez en exécutant le code sur machine.

##### 1<sup>er</sup> programme :

```
#include <stdio.h>
int main ()
{
    int n=10, p=5, q=10, r;
```

```

r = n = (p = q);
printf ("A : n = %d p = %d q= %d r = %d \n", n, p, q, r);
n = p = q =5;    n+=p+=q;
printf ("B : n = %d p = %d q= %d \n", n, p, q);
q = n<p ? n++ : p++;
printf ("C : n = %d p = %d q= %d \n", n, p, q);
q = n>p ? n++ : q++;
printf ("D : n = %d p = %d q= %d \n", n, p, q);
}

```

## 2<sup>ème</sup> programme :

```

#include <stdio.h>
int main ()
{
    int n,p,q;
    n=5; p=2; q= n++>p || p++ !=3;
    printf ("A : n = %d p = %d q= %d \n", n, p, q);
    n=5; p=2; q= n++ < p || p++ != 3;
    printf ("B : n = %d p = %d q= %d \n", n, p, q);
    n=5;p=2; q= ++n == 3 && ++p == 3;
    printf ("C : n = %d p = %d q= %d \n", n, p, q);
    n=5; p=2; q = ++n == 6 && ++p == 3;
    printf ("D : n = %d p = %d q= %d \n", n, p, q);
}

```

## Exercice 4

Si x vaut 3.456 et n 345 donner l’affichage à l’écran suite à l’exécution des instructions suivantes :

```

printf ("trois valeurs de x sont %4.1f*%5.2f*%.3f\n", x,x,x) ;
printf ("trois valeurs de n sont %4d*%5d*%d\n ", n,n,n) ;

```

## Exercice 5

Ecrire les instructions C qui correspondent aux formules suivantes. Choisir les identificateurs nécessaires appropriés, considérer que toutes les variables sont réelles et définir des constantes si nécessaires :

a) La période t d’un pendule de longueur h est donné

$$t = 2 \pi * (h/g)^{1/2}$$

où g est la constante gravitationnelle (981 cm /sec<sup>2</sup>)

b) La force attractive F entre deux corps de masse m1 et m2 séparés par une distance r

$$F = Gm_1m_2 /r^2$$

Où G = 6.673\*10<sup>-8</sup> cm<sup>3</sup>/g sec<sup>2</sup> est la constante gravitationnelle universelle

## Exercice 6 :

En C, les caractères sont codés par leur code ASCII compris entre 0 et 255, c’est-à-dire par un nombre entier. Ecrire un programme qui, donnant un caractère numérique, renvoie la valeur entière qu’il représente (et pas son code ASCII).

## TD 2bis de langage C

### **Exercice 7 :**

Ecrire un programme qui permet de calculer un prix TTC à partir d'un prix hors-taxé saisi au clavier, le taux de TVA étant fixé à 19,6%

### **Exercice 8 :**

Ecrire un programme qui calcule la moyenne de 2 puis 3 valeurs entières. On utilisera une seule variable pour la saisie des valeurs.

### **Exercice 9 :**

Ecrire un programme qui lit 2 variables de type double, les affiche, échange leurs contenus et affiche de nouveau leurs valeurs.

### **Exercice 10 :**

Ecrire un programme capable de convertir une donnée en heures, minutes et secondes en un nombre de secondes, et réciproquement (sans utiliser de conditionnelles).