

TD 1 - Manipulation des concepts élémentaires et priorité des opérateurs

Techniques de Programmation

Objectif

- Ce premier TP introductif à la programmation en C va vous familiariser avec :
- la syntaxe du langage et la mise en oeuvre du compilateur ;
 - les priorités des opérateurs.

1 Concepts élémentaires

1.1 Pour recommencer en douceur

Implémenter le programme HelloWorld.c vu en cours. Compilez-le et créer un programme HelloWorld.

1.2 Histoire de types

Implémenter un programme qui saisit deux entiers et en affiche le produit. Utiliser pour cela des entiers de type `short`. Puis des entiers de type `int`. Puis des entiers de type `long`. Tester le programme aux extrêmes. Que remarquez vous ?

Modifier le programme pour qu'il fonctionne avec des réels.

Quel est le nombre d'octets codant les types `short`, `int` et `long` sur ce compilateur ?

1.3 Et si...

a) Implémenter un programme qui saisit un nombre et vérifie qu'il est compris entre 0 et 20. Modifier le programme afin de vérifier si une série de nombre est comprise entre 0 et 20. Modifier le programme pour calculer la moyenne de cette série de nombre. Quelles sont vos remarques sur cet énoncé ?

b) Implémenter un programme qui saisit un élément au clavier et affiche s'il s'agit d'une minuscule, d'une majuscule, d'un chiffre ou d'un autre type de caractère. Répéter la saisie tant que l'élément saisi n'est pas <Esc>.

1.4 Switch ou pas switch

Implémenter un programme, en utilisant une imbrication de `if` dans un premier temps, puis en utilisant un `switch` ensuite, qui affiche si un étudiant est reçu à un examen en fonction de la note qui lui est donnée. Dans le cas où il est reçu, le programme affiche la mention. Les mentions sont *passable* pour une note comprise entre 10 et 12, *assez-bien* pour une note comprise entre 12 et 14, *bien* pour une note comprise entre 14 et 16, et *très-bien* pour une note comprise entre 16 et 20. Que remarquez vous ? Comment faut-il programmer les mentions pour qu'on puisse les changer facilement ?

2 Les opérateurs

Les exercices suivant doivent être réalisés sans machine.

2.1 Opérateurs arithmétiques de base

Que donne comme résultat le programme suivant ?

```
main(){
    int x;

    x = - 3 + 4 * 5 - 6; printf("%d\n", x);
    x = 3 + 4 % 5 - 6; printf("%d\n", x);
    x = - 3 * 4 % - 6 / 5; printf("%d\n", x);
    x = ( 7 + 6 ) % 5 / 2 ; printf("%d\n", x);
}
```

2.2 Opérateurs d'affectation

Que donne comme résultat le programme suivant ?

```
main(){
    int x=2, y, z;

    x *= 3 + 2; printf("%d\n", x);
    x *= y = z = 4; printf("%d\n", x);
    x = y == Z; printf("%d\n", x);
    x == ( y = z); printf("%d\n", x);
}
```

2.3 Opérateurs logiques et d'incrément

Que donne comme résultat le programme suivant ?

```
main(){
    int x, y, z;

    x = 2; y = 1; z = 0;
    x = x && y || z; printf("%d\n", x);
    cout << (x || ! y && z);

    x = y = 1 ;
    z = x ++ - 1; printf("%d\n", x); printf("%d\n", z);
    z += -x ++ + ++ y; printf("%d\n", x); printf("%d\n", y); printf("%d\n", z);
    z = x / ++ x; printf("%d\n", z);
}
```

2.4 Opérateurs relationnels et conditionnels

Que donne comme résultat le programme suivant ?

```
main(){
    int x=1, y=1, z=1,r=0;

    x += y += z;
    r = (x < y ? y : x );
    printf ("r=%d\n\n",r);

    r = (x < y ? x ++ : y ++);
    printf ("r=%d\n",r); printf ("x=%d\n",x); printf ("y=%d\n\n",y);

    r = (z += x < y ? x ++ : y ++);
    printf ("r=%d\n",r); printf ("z=%d\n",z);
    printf ("y=%d\n\n",y);

    x=3; y = z = 4;
    r = ((z >= y >= x) ? 1 : 0);
    printf ("r=%d\n\n",r);
    r = (z >= y && y >= x );
    printf ("r=%d\n\n",r);
}
```