

Langage C – TD2 – Expressions et opérateurs

J. SEINTURIER / Université de Toulon (julien.seinturier@univ-tln.fr / <http://www.seinturier.fr>)

1. Expressions arithmétiques

Exercice 1.1.

Citer les cinq opérateurs arithmétiques du langage C. Préciser leur sémantique et leur nature (unaire, binaire, ...).

Exercice 1.2.

Ecrire de 4 façons différentes une expression qui affecte à une variable x sa valeur plus 1 (incréméntation de 1).

Exercice 1.3.

Soit une variable x de valeur 1. Que retournent les expressions x--, --x et x -= 1 ? Quelle sera la valeur de x après chaque expression (les expressions sont évaluées indépendamment) ?

Exercice 1.4.

Soit un opérateur arithmétique o (pouvant être +, -, *, / ou %) et soit les types char, short, int, float et double. De quel type est l'opérateur arithmétique o pour chaque couple de type possible ? En déduire une règle de conversion de type pour les opérateurs arithmétiques.

2. Expressions logiques

Exercice 2.1.

Quel sont les équivalents numériques des valeurs booléennes VRAI et FAUX en langage C ?

Exercice 2.2.

Citer les opérateurs logiques du langage C et expliciter leur sémantique.

Exercice 2.3.

Donner les tables de valeur des opérateurs logiques du langage C.

Exercice 2.4.

Soit a et b deux variables. Ecrire en langage C l'expression logique « non (a et b) ou non b et non a ». Quel problème peut se poser lors de l'évaluation de l'expression ?

3. Expressions relationnelles

Exercice 3.1.

Citer les six opérateurs relationnels du langage C et expliciter leur sémantique et leurs valeurs de retour.

Exercice 3.2.

Soit trois variables numériques a, b et c. Ecrire l'expression « a est différent de b et b strictement supérieur à c ou a inférieur ou égal à c » en langage C. Quel problème peut se poser lors de l'évaluation de l'expression ?

4. Priorité des opérateurs

Exercice 4.1.

Comment le langage C rend-il déterministe l'évaluation d'expression contenant plusieurs opérateurs ?

Exercice 4.2.

Donner la table des priorités des opérateurs arithmétiques, logiques et relationnels du langage C.

Exercice 4.3.

Soit l'expression suivante :

```
++a || (++b > ++c && (++d*++e))
```

Si avant l'expression a = 1, b = 1, c = 1, d = 1 et e = 1, les valeurs des variables après l'expression sont a = 2, b = 1, c = 1, d = 1 et e = 1. Que s'est-il passé durant l'exécution ?

Exercice 4.4.

Soit l'expression suivante.

```
++x || (++y > z && (y*++z))
```

Ecrire l'arbre d'exécution de l'expression.

Numéroter les nœuds de l'arbre dans l'ordre de l'évaluation de l'instruction.

Faire apparaître les optimisations (en retirant de l'arbre les nœuds inutilisés) si les valeurs des variables avant expression sont :

- x = 1, y = 1 et z = 1
- x = -1, y = 1 et z = 3
- x = -1, y = 1 et z = 0

Exercice 4.5

On considère la séquence d'instructions suivante :

```
int a = 5, b = 3, c = 2;
```

```
int x = 0, y = 0, z = 1;
```

```
a = b;  
x || (y && z);  
a & b;  
a + b - c;  
(x == y) || (x != z);  
(a | b) & c;  
(a ^ b) || c;  
(a * c) / b;  
(a < b) != ( ! (b >= c) );
```

Ecrire un tableau d'exécution pour la séquence complète.

5. Entrées / sorties simples

Exercice 5.1.

Expliquer brièvement le fonctionnement de la fonction d'affichage `printf` du langage C. Décrire en particulier l'affichage de valeurs (variables, expressions).

Exercice 5.2.

Quel est l'affichage des 2 instructions suivantes (exécutées à la suite) :

```
printf("Hello");
printf("World");
```

Comment modifier ces instructions pour afficher :

```
Hello
World
2022
```

Exercice 5.3.

Qu'affiche le programme suivant :

```
#include <stdio.h>

int main() {
    char c = 'A';
    int i = 66;
    float f = 67.6;

    printf("char avec %%c : %c\n", c);
    printf("char avec %%d : %d\n", c);
    printf("char avec %%f : %f\n", c);
    printf("char avec %%lf: %lf\n", c);

    printf("\n");
    printf("int avec %%c : %c\n", i);
    printf("int avec %%d : %d\n", i);
    printf("int avec %%f : %f\n", i);
    printf("int avec %%lf: %lf\n", i);

    printf("\n");
    printf("float avec %%c : %c\n", f);
    printf("float avec %%d : %d\n", f);
    printf("float avec %%f : %f\n", f);
    printf("float avec %%lf: %lf\n", f);

    return 0;
}
```

Proposer un modèle de comportement des arguments de mise en forme %.

Exercice 5.4.

Expliquer brièvement le fonctionnement de la fonction de saisie `scanf` du langage C. Décrire en particulier le formatage des entrées.

Exercice 5.5.

Ecrire un programme qui lit deux entiers au clavier et affiche leur somme et leur différence.

Exercice 5.6.

Ecrire un programme qui lit un `int` et un `float` au clavier et affiche leur somme et leur différence.

6. Pour aller plus loin

Exercice 6.1.

Soient les déclarations suivantes :

```
int n=10; p=4;
long q=2;
float x=1.75;
```

Donner le type et la valeur de chacune des expressions suivantes :

- | | |
|-------------|----------------------|
| a) $n+q$ | f) $n>q$ |
| b) $n+x$ | g) $q+3*(n>p)$ |
| c) $n\%p+q$ | h) $(q-2)\&\&(n-10)$ |
| d) $n<p$ | i) $x*(q==2)$ |
| e) $n>=p$ | j) $x*(q==5)$ |

Exercice 6.2. (Minute environnement)

Quel peut être l'impact écologique (empreinte, charge, ...) d'un programme pour lesquels les types de variable sont surdimensionnés ou inadaptés ?