

Langage C – TD1 – Généralités, variables et types simples

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

1. Généralités

Exercice 1.1.

Décrire le processus de création d'un programme en langage C. Utiliser pour cela les notions de problème, formalisation, algorithme, codage, compilation et exécution (vous pouvez faire un schéma).

Exercice 1.2.

Soit le fichier prog.c contenant le code d'un programme écrit en langage C. Donner la commande permettant de compiler prog.c en un exécutable nommé prog.

Exercice 1.3.

En langage C, qu'est-ce qu'une instruction précédée d'un #. Comment fonctionne ce type d'instruction ?

Exercice 1.4.

Quel est le fonctionnement de l'instruction #include lors de la compilation ? Quel est le fonctionnement de l'instruction #define ?

2. Variables

Exercice 2.1.

En langage C, qu'est-ce qu'une variable ? Qu'est-ce qu'un type ?

Exercice 2.2.

Quels sont les types numériques disponibles en langage C ? Précisez leur plage de valeurs et leur taille d'encodage.

Exercice 2.3.

Donner le type et le nom de la variable destinée à recevoir les informations suivantes :

- votre âge
- la surface d'une parcelle cadastrale en m²
- l'âge moyen des personnes de votre famille
- Le nombre d'étoiles dans l'univers

Exercice 2.4.

Expliquer ce que font chacune des lignes suivantes :

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

Exercice 2.5.

Déclarer une constante pour pi=3,14159. Déclarer ensuite une variable puis l'initialiser à pi.

Exercice 2.6.

Soient les trois instructions suivantes :

```
int n = 5, p;
int q = 5.502;
float x = q;
```

Quelle sont les valeurs des variables n, p, q et x ?

Exercice 2.7.

Indiquer pourquoi les lignes suivantes sont incorrectes et les corriger :

1. -2 = x;
2. (x -1) = 2;
3. x = x+1
4. y =+ 5;

Exercice 2.8.

Parmi les déclarations suivantes, lesquelles sont correctes et lesquelles sont fausses :

1. float var, var1, VAR2;
2. real x,y,z;
3. int void, main;
4. char RS-232;
5. char 1carac, 2carac;

3. Types simples

Exercice 3.1.

Les variables de type char sont codées sur 1 octet. Le nombre entier qui les représente correspond au codage ASCII du caractère correspondant. Les codes ASCII principaux sont : '0' : 48, '1' : 49 etc., 'A' : 65, 'B' : 66 etc., 'a' : 97, 'b' : 98 etc.

Soit la séquence d'instructions suivante :

```
char c1, c2, c3;
c1 = '0'; c2 = 'A'; c3 = 'a';
```

Ecrire les octets représentant le stockage des valeurs : c1, c2, c3, c1+2, c2+c3.

Exercice 3.2.

1. Donner la représentation de l'entier 18 dans le type short (2 octets).

2. Pour tout nombre x positif représenté en base 2, la représentation en base 2 de son opposé -x se construit en allant du bit de poids faible vers le bit de poids fort : on laisse les premiers zéros puis le premier 1 inchangés, on intervertit ensuite tous les autres chiffres. (Remarque, pour le zéro (0000 0000) il n'y a pas de premier un et donc on ne fait rien. En déduire la représentation de -18 en base 2.

3. On rappelle que le type char permet en fait de représenter des entiers signés sur un seul octet ; quel est l'intervalle des entiers représentables dans ce type ?

4. Soit un programme en C :

```
#include <stdio.h>
int main()
{
    char c = 'a';
    char d = c+c;
}
```

Que valent c et d à la fin du programme ? Pourquoi ?