Pratique du C :
Expressions et nombres

Licence Informatique — Université Lille
Pour toutes remarques : Alexandre.Sedoglavic@univ-lille1.fr

Semestre 4 — 2017-2018

Les principaux opérateurs et leurs priorités sont :

16	`()` `[]` `->` `.`	G
15	`++` `--` (postfixé)	D
14	`!` `~` `++` `--` (préfixé) `-` (unaire)	D
	`*` (indirection) `&` (adresse) `sizeof`	D
13	`*` (multiplication) `/` `%`	G
12	`+` `-`	G
11	`<<` `>>`	G
10	`<` `<=` `>` `>=`	G
9	`==` `!=`	G
8	`&` (et bit à bit)	G
7	`^`	G
6	`|`	G
5	`&&`	G
4	`||`	G
3	`?:`	D
2	`=` `+=` `-=` `*=` `/=` `%=` `>>=` `<<=` `&=` `^=` `|=`	D
1	`,`	G

Exercice 1 — Les expressions et leurs évaluations.

En supposant que :

$$A = 20 \quad B = 5 \quad C = -10 \quad D = 2 \quad X = 12 \quad Y = 15$$

évaluer les expressions valides parmi les expressions suivantes :

$5 * X + 2 * 3 * B/4$	$A == B = 5$	$A+ = X + 2$	$A! = C* = -D$
$A\% = D + +$	$A\% = + + D$	$(X + +) * A + C$	$A = 6 * D == X?B : Y$
$!(X - D + C)\| \| + +D$	$A\&\&B\|\|!0\&\&C\&\&!D$	$C = 12 - -$	`(1<<(2<<1))==((1<<2)<<1)`

En utilisant le parenthésage, corriger les expressions invalides.

Exercice 2 — Type atomique et conversion implicite.

En supposant faites les déclaration :

```
long int A = 15 ;
char B = 'A' ;
short int C = 10 ;
```

évaluer et donner le type des expressions valides suivantes :

$B + 1$	$B + A$	$B + C$
$3 * B + 2 * B$	$2 * B + (A + 10)/C$	$2 * B + (A + 10.0)/C$
$B = 2 * B + (A + 10.0)/C$	$C = 666$	$B* = 3.14$

Exercice 3 — Entrée élémentaire.

Donnez la définition d'une fonction de prototype :

```
unsigned int str2uint(void) ;
```

1

qui *calcule* la valeur d'un entier — non signé — entré sur `stdin` sous forme d'une suite de caractères (terminée par un caractère autre qu'un chiffre).

L'arithmétique *classique* peut s'appliquer sur le type `char`. La valeur numérique d'un caractère est le code ASCII de ce caractère (48 pour `'0'`, 49 pour `'1'`,etc.).

Exercice 4 — Sortie élémentaire.

Donnez la définition d'une fonction d'identificateur `AfficheEntier` qui prends un argument de type `unsigned int`, ne retourne rien et affiche sur sa sortie standard la conversion en chaîne de caractères de cet entier. On utilisera la fonction de prototype `int putchar(int)` pour l'affichage.

Exercice 5 — Affichage formatté.

Donnez la définition d'une fonction de prototype :

```
void affiche_ligne_xxd(char tab[]) ;
```

qui affiche la chaîne de caractères codée par le tableau `tab` en respectant le format suivant :

- la première colonne de 7 caractères représente l'offset du premier caractère du paquet de 16 caractères considérés du fichier ;
- la seconde colonne comporte 32 caractères groupés par parquet de 4. Elle indique le codage hexadécimal des caractères considérés ;
- la dernière colonnes de 16 caractères correspond à l'affichage de ces derniers (les caractères non imprimables — e.g. retour charriot — sont remplacés par un point).

Par exemple, appliquée à la chaîne `La vie est belle.`", la fonction affiche :

```
0000000: 4C61 2076 6965 2065 7374 2062 656C 6C65 La vie est belle
0000010: 2E0A                                     ..
```