

# Variables et expressions

## Notion de variable

- Les variables servent à stocker les données manipulées par le programme
- Les variables sont nommées par des identificateurs alphanumériques.

## Noms de variables

- Symboles autorisés :  
a-z A-Z 0-9 \_
- Commence par une lettre
  - Pas de symbol \_ au début, car cette notation est réservée aux variables système
- 31 premiers caractères significatifs
- Exemples :  
un  
Paul  
R2D2  
Nouvelle\_Variable
- Les mots clés sont réservés  
if else int float

## Types de données

- Types simples :
  - char : codé sur un octet, utilisé pour représenter le code ASCII d'un caractère
  - int : entier de base de l'architecture
  - float : réel simple précision
  - double : réel double précision
- Modificateurs :
  - short, long : précisent la taille d'un entier (2 ou 8 octets)
  - unsigned : précise que l'entier est naturel (>0). S'applique également aux caractères.
- Exemple :  
unsigned long int compteur

## Valeurs (1)

- Valeurs entières :
 

1234	décimal	(1234)
01234	octal	(668)
0x12345	hexadécimal	(4660)

Le suffixe l ou L indique une valeur longue

Le suffixe U indique une valeur non signée.

- Valeurs réelles :
 

1.234	notation fixe
1.234 e-10	notation scientifique

## Valeurs (2)

- Caractères :
 

c'est le code ASCII du caractère (entier entre 0 et 127)

'a'	code ASCII du a
'\n'	code ASCII du line feed
'\t'	code ASCII du tabulateur
'\001'	code binaire 1 (octal)
'\x0f'	code binaire 15 (hexadécimal)
- Chaînes :
 

Tableau de caractères
” Ceci est une chaîne”
- Valeurs énumérées :
 

enum normand {OUI,NON,PEUT_ETRE}
----------------------------------

## Déclarations

- Syntaxe :  

```
type liste_de_variables;
```
- Exemples :  

```
int lower;
int lower,upper,step;
char c,buf[1000],esc='\';
unsigned char x;
const float pi = 3.141592;
const char message[] = "alerte !!!";
int strlen(const char[]);
```

MIAGE - Université de Bordeaux

7

## Opérations arithmétiques

- Quatre opérations de base :
  - +, -, \*, /
  - Opérations typées : le type des opérandes détermine le type de l'opération
- Opérateur %  
 reste de la division (modulo)
- Exemple :  

```
printf("%d",year);
if((year%4==0 && year%100!=0) || year%400==0)
    printf("est");
else
    printf("n'est pas ");
printf("bissextille\n");
```

MIAGE - Université de Bordeaux

8

## Opérateurs logiques

- Comparaisons :
  - > supérieur
  - < inférieur
  - >= supérieur ou égal
  - <= inférieur ou égal
  - == égalité
  - != différence
- Combinaisons :
  - ! contraire
  - && et logique
  - || ou logique
  - Remarques :
    - Le résultat de l'opération prend une valeur égale à 0 en cas d'échec (faux) et différent de 0 en cas de succès (vrai)
    - Écrire if(valide) plutôt que if(valide != 0)

## Conversions de type

- Conversions implicites de type :  
en cas de conflit entre opérandes, on convertit dans le type le plus fort avec  
char < short <= int <= long < float < double
- Conversions explicites de type : (casting)  
(type) expression  
sert à forcer une conversion non implicite.
- Exemples :  
sqrt((double)n);  
a = (unsigned char)x;

## Opérateurs incrémentaux

- Opérateurs unaires ++ et –
  - Prés incrémentation :  
incrémente la variable avant d'évaluer l'expression dans laquelle elle figure.
  - Post incrémentation :  
incrémente la variable après avoir évalué l'expression dans laquelle elle figure.
- Exemples :
  - `s[++i] = c` signifie ++i; `s[i] = c;`
  - `s[i--] = c` signifie `s[i] = c; i--;`

MIAGE - Université de Bordeaux

11

## Opérateurs binaires

&	et bit à bit
	ou bit à bit
^	ou exclusif bit à bit
<<	décalage à gauche
>>	décalage à droite
~	complément à 1

- Exemples :
 

```
x = x & 0177;
unsigned getbits(unsigned x, int p, int n){
    return (x>>(p+1-n)) & ~(~0<<n);
}
```

MIAGE - Université de Bordeaux

12

## Opérations d'affectation

- L'affectation est une opération évaluable  
 $a=b=c$  est interprété comme  $a=(b=c)$
- Les expressions de la forme  $a = a+x$  peuvent s'écrire  $a += x;$
- Applicables aux opérateurs  $+ - * / << >> \& \wedge |$
- Exemple :  
`t[x[i][j],y[k][l]] += 25;`

MIAGE - Université de Bordeaux

13

## Expressions conditionnelles

- Expression qui peut prendre deux valeurs différentes en fonction d'une troisième valeur.  
`if(a>b)`  
`z = a;`                    peut s'écrire : `z = (a>b)?a:b;`  
`else`  
`z = b;`
- Exemple :  
`printf("il y a %d erreur%s\n",n,(n==1)?"", "s");`

MIAGE - Université de Bordeaux

14

## Effets de bord

- Le langage C ne précise pas dans quel ordre sont évaluées les expressions. Certaines écritures sont dépendantes du compilateur :

```
x = depiler() - depiler();
```

dans cette expression, selon l'ordre d'évaluation des expressions, le résultat peut varier.

- Autre exemple :

```
printf("%d, %d\n", n, ++n*power(2,n));
```

## Exercices

- Écrire, compiler et exécuter les programmes qui sont les solutions des exercices suivants :
  - Afficher le pourcentage de blancs d'un fichier. On écrira un programme qui fait ses lectures au clavier.
  - Supprimer les blancs non significatifs d'un fichier, c'est-à-dire, ceux qui se trouvent en fin de ligne.
  - Étendre les tabulations d'un fichier, c'est-à-dire, remplacer les caractères de tabulation par un nombre suffisant de blancs pour simuler la tabulation.