

Introduction au C

University of Geneva
www.miralab.ch

Rappel :

- Généralités
 - Programmation
 - Compilation
 - Caractéristiques du C
- Pt d'entrée d'1 prog. C
- Commentaires
- Instruction élémentaire, bloc
- Constantes
 - Nombre, caractères, chaînes de caractères
- Variable
 - Affectation & déclaration
- Types
 - Int, char, float, double, string
- Expressions arithmétiques
 - Règles d'évaluation

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

- Booléens
- Écriture/Lecture de données
- Définition de type
- Pointeurs

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Booléens

University of Geneva
www.miralab.ch

Variable booléenne en C

Pas de variables booléennes en C

Toute valeur **=0** correspond à **FAUX**

Toute valeur **!=0** correspond à **VRAI**

Résultat d'une opération logique vaut **0** ou **1**

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Variable booléenne en C++

Type : **bool**

Constantes : true, false

Compatible avec C

false ⇔ =0

true ⇔ !=0

// déclaration

bool waiting;

bool ready;

// décl. + affectation

bool correct = true;

// affectation

ready = false;

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Opérateurs logiques

3 opérateurs logiques:

« et » **&&**
« ou » **||**
« non » **!**

$\text{non_q} = 1 \Leftrightarrow q = 0$
 $= 0 \Leftrightarrow q = 1$
 $\text{p_et_q} = 1 \Leftrightarrow \text{p et q} \neq 0$
 $\text{p_ou_q} = 0 \Leftrightarrow \text{p et q} = 0$

```
int p, q, p_et_q;
int p_ou_q;
int non_q

p_et_q = p && q;
p_ou_q = p || q;
non_q = !q;
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Expressions booléennes

Contenant plusieurs opérateurs logiques:

exemple

```
bool active, initialisee, finie, prete;
[...] // initialisation
active = (prete && initialisee) || !finie;
```

$\text{active} = \text{true} \Leftrightarrow (\text{prete} \text{ et } \text{initialisee}) = \text{true}$
ou
 $\text{finie} = \text{false}$

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Priorité des opérateurs

Ordre de priorité : **! → && → ||**

Si égalité de priorité
évaluation de gauche à droite

Expressions **()** évaluées en 1^{er}

Mêmes règles d'évaluation que les expressions arithmétiques

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Exemple

```
bool p = true, q = true;
bool r, s = false;
r = !p || q && s;
```

est évaluée de la sorte :

$r = \text{!true} || \text{true} \&\& \text{false}$
 $= \text{false} || \text{false}$
 $= \text{false}$

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Exemple (utilisation de parenthèses)

Pour préciser l'ordre d'évaluation
 $\text{bool } p = \text{false}, q = \text{false}, r, s = \text{true};$
 $r = \text{!} ((p || q) \&\& s);$

est évaluée de la sorte :

$r = \text{!} ((\text{false} || \text{false}) \&\& \text{true})$
 $= \text{!} (\text{false} \&\& \text{true})$
 $= \text{! false}$
 $= \text{true}$

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Les opérateurs relationnels

opérateur en C	symbole usuel	signification
>	>	plus grand que
>=	≥	plus grand ou égal
<	<	plus petit que
<=	≤	plus petit ou égal
==	=	égal
!=	≠	différent

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Priorité de ces opérateurs

Inférieure aux opérateurs arithmétiques
rappel : +, -, *, /, %

Les opérateurs >, >=, <, <= ont une priorité
supérieure à == et !=

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Expressions booléennes simples

Doivent être de **même** type.

Exemple

```
int quantite = 10;
bool charbon, Taxable, celibataire = true, voyelle;
float revenu = 24000.0f;
char lettre = 'G';
```

```
charbon = quantite > 0;
taxable = (revenu >= 12000.0f) && celibataire;
voyelle = (lettre == 'A' || lettre == 'E' || lettre == 'I');
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

« c type casting »

= non respect de la correspondance des
types
conversions de type implicite

Résultat peut s'avérer très différent de celui
escompté ...

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

« C type casting » implicite et explicite

```
int quantite = 0;
bool charbon, taxable, celibataire = false;
float revenu = 4000.0f;
char lettre = 'w';
```

```
// c type casting implicite
charbon = quantite > revenu;
taxable = (revenu > lettre) && celibataire;
```

```
// c type casting explicite
revenu = (float)quantite;
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Écriture/Lecture de données

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Écriture sur la console

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Écriture sur la console (rappel)

Utiliser la librairie iostream

```
#include <iostream>
using namespace std;
```

Fonction

```
cout << " ... " << endl;
```

redirection du flux

option

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Composition

- En C
 - printf("`La valeur i = %d et j = %f`, i, j);
- En C++
 - Cout << "`La valeur i = "` << i << "` et j = "` << j << endl;

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Lecture de données

En C

Fonction fscanf

```
fscanf(stdin, "%d %d", &i, &j);
```

En C++

Fonction **cin** (avec <iostream>)

```
cin >> i >> j;
```

Exemple :

```
int i,j;
cin >> i >> j;           // en C++ avec <iostream>
fscanf(stdin, "%d %d", &i, &j); // en C pur
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Lecture d'une données ou plusieurs

```
int i, j;
```

```
cin >> i;
```

```
cin >> j;
```

#

```
int i,j;
```

```
cin >> i >> j;
```

Lecture successive de 2 valeurs

Lecture simultanée de 2 valeurs

Entrée
15
10

Entrée
15 10

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Nouveaux types

Définition de nouveaux types

Syntaxe

```
typedef type_existant nom_type;
```

Exemple

```
typedef long int grand_entier_t;
```

```
grand_entier_t a,b;
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Type par énumération

Définit une liste d'identificateurs ou labels
 Associe valeur **entière** aux labels
 Par défaut, le 1^{er} label = 0, les suivants sont
 incrémentés de 1 en 1

Syntaxe

```
typedef enum { liste_def_label } nom_type;
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Exemple

```
typedef enum {
    lundi, mardi, mercredi, jeudi, vendredi,
    samedi, dimanche, abstinance
} jour_t;
```

```
jour_t aujourd'hui = mercredi;
```

```
int dimanche; // erreur ce n'est pas un 'int' au sens commun
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Définition explicite équivalente

typedef enum

```
{ lundi,
  mardi,
  mercredi,
  jeudi,
  vendredi,
  samedi,
  dimanche
} jour_t;
```

↔

typedef enum

```
{ lundi=0,
  mardi=1,
  mercredi=2,
  jeudi=3,
  vendredi=4,
  samedi=5,
  dimanche=6
} jour_t;
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Indexage de 1 à 7

typedef enum

```
{ lundi=1,
  mardi,
  mercredi,
  jeudi,
  vendredi,
  samedi,
  dimanche
} jour_t;
```

↔

typedef enum

```
{ lundi=1,
  mardi=2,
  mercredi=3,
  jeudi=4,
  vendredi=5,
  samedi=6,
  dimanche=7
} jour_t;
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Affectation contrainte

variable de type *jour_t* peut prendre
 n'importe laquelle des 7 valeurs
 prédéfinies **et seulement** celles-ci.

```
jour_t aujourd'hui = mercredi;
aujourd'hui = mardi;
// aujourd'hui = 400;
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Attention !!!

Le type de la variable et celui de la valeur
 qu'on lui affecte doivent être les mêmes

```
jour_t aujourd'hui = mercredi;
int Saturday = 5;
aujourd'hui = Saturday; // erreur
aujourd'hui = 5; // erreur
aujourd'hui = (jour_t) 5; // c-casting
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Attention !!! le retour

31

Valeurs de la liste sont des entiers
 Pas de chaînes de caractères
 Impossible de lire ou d'afficher comme chaîne
 de caractères

```
jour_t aujourd'hui = mercredi;
cout << "jour = " << aujourd'hui << endl;
```

Affiche
 jour = 2
 et non pas
 jour = mercredi !!!!

University of Geneva
www.miralab.chLCI
Introduction au langage C**Attention !!!! la revanche**

32

Un même label ne pas figurer dans deux
enum différents

```
typedef enum {
    vert = 60, jaune = 70, orange = 80, bleu = 90
} couleur_t;

typedef enum {
    cerise = 60, citron = 70, orange = 80, fraise = 50
} fruits_t
```

ERREUR !

University of Geneva
www.miralab.chLCI
Introduction au langage C**Les pointeurs**University of Geneva
www.miralab.ch**Les pointeurs**

34

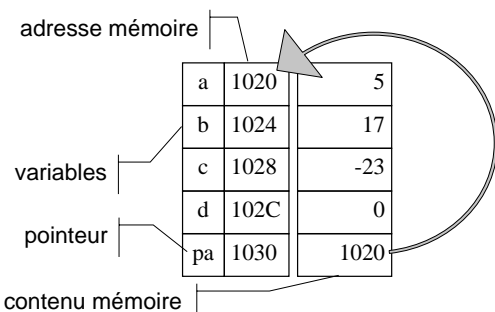
Pointeur = variable contenant l'adresse
 mémoire d'une autre variable

Une variable a 2 attributs

- la valeur qui lui est associée
- l'endroit de la mémoire où elle est rangée
 c'est l'adresse (mémoire) de la variable

University of Geneva
www.miralab.chLCI
Introduction au langage C**Adresse et contenu mémoire**

35

University of Geneva
www.miralab.chLCI
Introduction au langage C**Convention (syntaxe)**

36

Pour une variable **a**
&a désigne son adresse

Si **pa** = pointeur sur **a**
***pa** désigne le contenu de la variable **a**

```
int a = 5;
int *pa; //pointeur vers un int
pa = &a;
cout << "La valeur de a vaut " << *pa << endl;
```

University of Geneva
www.miralab.chLCI
Introduction au langage C

Déclaration de pointeur et type

Déclaration d'un pointeur

Placer * entre son nom et le type

float *pa; //pointeur vers un float

Type du pointeur =

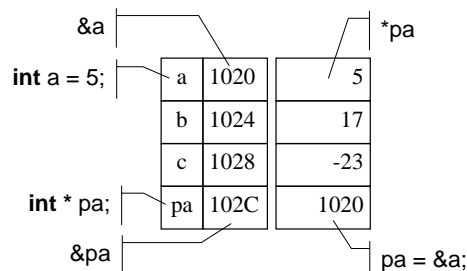
type de la variable dont il contient l'adresse

On ne peut pas affecter à un pointeur de type **int** l'adresse d'une variable de type **float**

University of Geneva
www.mirallab.ch

LCI
Introduction au langage C

Représentation mémoire



University of Geneva
www.mirallab.ch

LCI
Introduction au langage C

Opération sur l'adresse

Certaines opérations arithmétiques sur l'adresse des pointeurs sont possibles

int *pa; //pointeur vers un int

`pa += 2;`

`pa = pa - 1;`

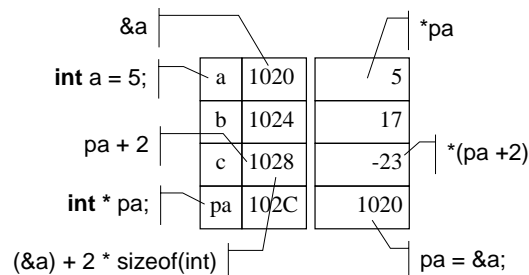
`pa--;`

ATTENTION !!! ON MODIFIE L'ADRESSE ICI, PAS LA VALEUR

University of Geneva
www.mirallab.ch

LCI
Introduction au langage C

Incrémentation de l'adresse



University of Geneva
www.mirallab.ch

LCI
Introduction au langage C

Exemple

int a = 5;

int *pa; //pointeur vers un int

float f = 1.23f;

float *pf; //pointeur vers un float

`pa = &a;` //pa pointe vers variable a

`*pa += 3;` /* identique à `a += 3`
donc `*pa = a = 8` */

`pa += 3;` /* attention `pa` pointe désormais 3 variables `int`
après `a` donc on ne sait PAS du tout vers quoi !!! */

`pf = &f;` // OK

`pa = pf;` // ILLEGAL

University of Geneva
www.mirallab.ch

LCI
Introduction au langage C

Constante NULL

Valeur particulière pour pointeur ne pointant nulle part, c.a.d. sur rien

float f = 1.23f;

//ptr vers un float initialisé à NULL

float *pf = NULL;

`cout << *pf << endl;` // Illégal : le pointeur ne pointe nulle part

`pf = &f;` // OK

`cout << *pf << endl;` // OK, affiche 1.23 sur la console

University of Geneva
www.mirallab.ch

LCI
Introduction au langage C

Initialisation des pointeurs

Comme pour les autres variables
lors de leurs déclarations (ou après)
à condition de ne pas référencer une variable avant
qu'elle n'ait été définie

```
float f = 1.23f;
float * pf = &f; // correct
```

```
int *pi = &q; // illégal
int qi = 8;
pi = &q; // correct
```

University of Geneva
www.mirab.ch

LCI
Introduction au langage C

Comparaison de pointeur

Opérateurs relationnels restent valides

`==, !=, <, >, <=, >=`

Les plus utiles sont l'égalité et la différence

```
bool isValid, isNull;
float revenu=24000.0f, * pf=NULL;
isNull = (pf == NULL);
pf = &revenu;
isValid = (pf != NULL);
```

University of Geneva
www.mirab.ch

LCI
Introduction au langage C

Erreurs communes

Oubli de l'initialisation de l'adresse mémoire
avant l'utilisation du pointeur.

```
float * pf, fval;
*pf = 10.0f; //erreur non-détectée à la compilation
pf ne correspond à aucune adresse mémoire
il faut d'abord initialiser pf
pf = &fval;
*pf = 10.0f; //OK
```

University of Geneva
www.mirab.ch

LCI
Introduction au langage C

Erreurs communes (suite)

Indirection illégale

```
char * pc1, * pc2, cval2 = '2';
pc2 = &cval2;
*pc1 = pc2; //erreur idem pb. précédent
pc1 ne correspond à aucune adresse mémoire
de plus, même si c'était le cas
char cval1 = '1';
pc1 = &cval1;
*pc1 = pc2; //i.e. cval1 = &cval2;
c.a.d. on affecte à cval1 la valeur de l'adresse de cval2 !
```

University of Geneva
www.mirab.ch

LCI
Introduction au langage C

Résumé

Booléens

En C avec les **int**

En C++ : **bool**

Opérateurs logiques

Priorités

Expression booléenne

Opérateur relationnel

Lecture de données

Définition de nouveau type

typedef

Liste de constantes typées

enum

Les Pointeurs

Adresse mémoire **&**

Contenu mémoire *****

NULL

University of Geneva
www.mirab.ch

LCI
Introduction au langage C