

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

Rappel (suite)

Booléens

En C avec les *int*
En C++ : *bool*

Définition de nouveau type
typedef

Opérateurs logiques
Priorités

Liste de constantes typées
enum

Expression booléenne

Les Pointeurs

Opérateur relationnel

Adresse mémoire &

Lecture de données

Contenu mémoire *
NULL

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Plan

- Bloc de contrôle et portée
- if() ... Else
- switch
- while et do ... while
- for

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Instructions de contrôle

University of Geneva
www.miralab.ch

Le bloc de contrôle

Délimité par des accolades { }

Portée de déclaration de variables

```
{ float f = 1.23f;
  int bloc1 = 1;
  {
    int bloc2;
    bloc2 = bloc1 * f; // OK
  }
  bloc2 = bloc1 * f; // illégal : bloc2 n'est
                    // + défini dans le bloc courant
}
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

L'instruction *if()* ... *else*

Syntaxe :

```
if (expression booléenne 1)
{ /*bloc1*/ }
else
{ /*bloc2*/ }
```

Teste une condition

Effectue **sélectivement** un bloc d'instructions

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

L'instruction *if()* ... *else* (suite)

```
if (expression booléenne)
{ /*bloc1*/ }
else
{ /*bloc2*/ }
```

{ /*bloc1*/ } est exécuté si

le résultat de **expression booléenne** est vrai

{ /*bloc2*/ } est exécuté si

le résultat de **expression booléenne** est faux

Rq :

else est optionnel \Rightarrow if(expr.bool) { /* bloc*/ }

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Instruction imbriquée

Syntaxe :

```
if (expression booléenne 1)
{ /*bloc1*/ }
else if (expression booléenne 2)
{ /*bloc2*/ }
else
{ /*bloc3*/ }
```

{ /*bloc1*/ } est exécuté si

le résultat de condition 1 est vrai

{ /*bloc2*/ } est exécuté si

le résultat de condition 1 est faux et

le résultat de condition 2 est vrai

{ /*bloc3*/ } est exécuté si

le résultat des conditions 1 et 2 est faux

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Exemples

```
if(couleur == 3)
radio = 1;
```

pour couleur = 3

\rightarrow couleur == 3 \Leftrightarrow true (1^{er})

radio = 1;

```
if(couleur == 3)
{
radio = 1;
```

\rightarrow couleur == 3 \Leftrightarrow true (2nd)

\rightarrow radio = 1;

```
else if(couleur <= 2)
{
radio = 2;
```

pour couleur = 2

\rightarrow couleur == 3 \Leftrightarrow false (2x)

\rightarrow couleur <= 2 \Leftrightarrow true

\rightarrow radio = 2;

```
else
```

```
cerr <<
"radio inconnue"
<< endl;
```

pour couleur = 5

\rightarrow couleur == 3 \Leftrightarrow false (2x)

\rightarrow couleur <= 2 \Leftrightarrow false

\rightarrow "radio inconnue"

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

If imbriqués

```
if(couleur <= 3)
if(couleur == 2)
radio = 2;
else
cerr << "radio inconnue"
<< endl;
```

\Leftrightarrow

```
if(couleur <= 3)
{
if(couleur == 2)
radio = 2;
else
{
cerr <<
"radio inconnue"
<< endl;
}
}
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

L'instruction *switch*

Syntaxe :

```
switch(variable_entière)
{
case valeur1 :
//liste d'instructions
break ;
case valeur2 :
//liste d'instructions
break ;

//...

default :
//liste d'instructions
break ;
}
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

L'instruction *switch*

Permet de réaliser un choix multiple

Une action associée aux valeurs de la variable de type entier *int*, *char*, *long int*, ...

L'instruction *break* peut être omise ou remplacée par *return()*

en l'absence de *break* / *return()*

⇒ les actions associées aux case suivants seront aussi exécutées !!

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Cas spécial *default*:

Facultatif

Emploi similaire aux 'case' mais sans valeur

Figure nécessairement **après** tous les 'case'

Est activé lorsque tous les 'case' échouent
quand la valeur ne correspond à aucun cas

Permet généralement d'indiquer
un problème, un cas non prévu
ou d'activer une action par défaut

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

```
float a = 0.0f;
int n;
switch(n)
{
    case 1 :
        a += 1.0f;
        break;
    case 2 :
        a += 2.0f/3.0f;
        break;
    case 3 :
        a /= 5.0f;
        break;
    default :
        cout << "cas non
répertorié" << endl;
        return false;
}
```



```
float a = 0.0f;
int n;
switch(n)
{
    case 1 :
        a += 1.0f;
    case 2 :
        a += 2.0f/3.0f;
    case 3 :
        a /= 5.0f;
        break;
    default :
        cout << "cas non
répertorié" << endl;
        return false;
}
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Équivalence logique

```
switch(n)
{
    case 1 :
        a += 1.0;
    case 2 :
        a += 2.0/3.0;
    case 3 :
        a /= 5.0;
        break;
    default :
        cout<<"erreur"<<endl;
        break;
}
```

```
if (n == 1)
{ a += 1.0; a += 2.0/3.0;
  a /= 5.0; }
else if (n == 2)
{ a += 2.0/3.0; a /= 5.0; }
else if (n == 3)
  a /= 5.0;
else
  cout<<"erreur"<<endl;
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Équivalence logique (suite)

```
switch(n)
{
    case 1 :
        a += 1.0;
        break;
    case 2 :
        a += 2.0/3.0;
        break;
    case 3 :
        a /= 5.0;
        break;
    default :
        cout<<"erreur"<<endl;
        break;
}
```



```
if (n == 1)
    a += 1.0;
else if (n == 2)
    a += 2.0/3.0;
else if (n == 3)
    a /= 5.0;
else
    cout<<"erreur"<<endl;
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Différence entre *if()* et *switch*

switch()

ne teste pas les conditions l'une après l'autre

fonctionne qu'avec des variables **entières**

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Exemple : grand magasin

on enregistre le prix d'achat par catégorie de marchandise

- 'a' articles ménagers, taxe : 6%
- 'b' vêtements, pas de taxe
- 'c' meubles, taxe : 6%
- 'd' outils, taxe : 8%
- 'e' parfums, taxe : 11%

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

```
int main()
{
    float somme = 0, prix;
    char categorie;
    cout<<"Tapez la catégorie, puis le prix"<<endl;
    cin >> categorie; //saisie de la catégorie

    while(categorie != 'f')
    {
        cin >> prix; //saisie du prix
        switch(categorie){
            case 'a':
                case 'c': somme += 1.06f * prix; break;
                case 'd': somme += 1.08f * prix; break;
                case 'e': somme += 1.11f * prix; break;
                case 'b': somme += prix; break;
                default: cerr << "mauvaise catégorie " << endl; break;
        } //fin du switch()
        cin >> categorie;
    } //fin de la boucle while()
    cout << endl;
    cout << "Montant total = " << std::fixed
        << std::setprecision(2) << somme << endl;
    //fin du main()
}
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Exécution

Tapez la catégorie, puis le prix

a 339.95
b 15.17
e 6.25
a 236
c 1133.5
b 142.15
b 137
d 12.45
f

Montant total = 2126.72

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

L'instruction *while*

Syntaxe :

```
while(expression booléenne)
{ /*bloc*/ }
```

ou

```
do { /*bloc*/ }
while (expression booléenne);
```

Le bloc d'instruction est exécuté
tant que la condition est vraie

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Exemple *while*

```
int couleur = 1;
while(couleur <= 3)
{
    couleur++;
}
// couleur vaut 4
cout << "couleur ="
    << couleur
    << endl;
```

couleur = 1
→ couleur <= 3 ⇔ true
couleur = 2;
→ couleur <= 3 ⇔ true
couleur = 3;
→ couleur <= 3 ⇔ true
couleur = 4;
→ couleur <= 3 ⇔ false
→ "couleur = 4"

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Exemple *do ... while*

```
int couleur = 1;
do
{
    couleur++;
}
while(couleur <= 3);
// couleur vaut 4
cout << "couleur ="
    << couleur
    << endl;
```

couleur = 1
couleur = 2;
→ couleur <= 3 ⇔ true
couleur = 3;
→ couleur <= 3 ⇔ true
couleur = 4;
→ couleur <= 3 ⇔ false
→ "couleur = 4"

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Différence *while* vs *do .. while*

while condition { /*bloc*/ }

bloc peut ne jamais être exécuté

```
int couleur = 4;
while(couleur <= 3)
{
    couleur++;
}
// couleur vaut 4
```

do { /*bloc*/ }

while condition ;

bloc est exécuté au moins une fois même si le premier test de *condition* est faux

```
int couleur = 4;
do
{
    couleur++;
} while(couleur <= 3);
// couleur vaut 5
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Exercice : suite de Fibonacci

$f(n) = f(n-1) + f(n-2)$, $n > 1$

$f(0) = f(1) = 1$

→ $f(2) = 1+1 = 2$, $f(3) = 2+1 = 3$, $f(4) = 5$, ...

Question

Pour quelle 1^{ère} valeur de n , $f(n) > 500$?

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

```
#include <iostream>
using namespace std;
int main()
{
    //f(1)    f(0)
    int N = 500, fn = 1, fn2 = 1, fn1;

    do
    {
        fn2 = fn1; //f(n-2)
        fn1 = fn; //f(n-1)
        fn = fn1 + fn2; //f(n)
    }
    while (fn <= N);
    cout << "1er terme de la suite de Fibonacci "
         << "> " << N << " vaut " << somme << endl;
}
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

L'instruction *for*

Syntaxe :

```
for (expression initiale;
     expression booléenne;
     expression d'itération)
{ /* bloc à exécuter */ }
```

Évalue l'expression initiale

Répète

- le test de l'expression booléenne
- l'exécution du bloc tant que la condition est vérifiée
- l'exécution d'itération après chaque bloc

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Correspondance *for* et *while*

```
for(expr. initiale;
    expr. booléenne;
    expr. d'itération)
{
    /* bloc for */
}

⇔

expr. initiale;
while(expr. booléenne)
{
    /* bloc for */
    expr. d'itération;
}
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Exemple : boucle de test

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

JOUR_t j, journée=mercredi;
for(j = lundi; j <= dimanche; j++)
{
    if (journée == j)
        cout << "c'est Mercredi" << endl;
}
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Exemple : somme des carrés

somme = $\sum n^2$, pour $n = 1 \dots 100$

Algorithme

```
somme = 0
boucle n=1 à 100
    calculer  $x = n^2$ 
    somme = somme + x
afficher somme
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

```
int main( )
{
    int n, X, somme;
    somme = 0;

    for (n = 1;           //initialisations
        n <= 100;        //condition
        n ++;            //itérations
    )
    {
        somme = somme + (n * n);
    }
    cout << "La somme finale = " << somme << endl;
}
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Résumé

Bloc de contrôle { }
Portée de variable

Instructions de contrôle

```
if() ... else ...
switch()
case:
...
break;
while() && do ... while()
for() ...
```

University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Exercice 1 : for

Faire un programme qui calcule la moyenne de N notes saisies au clavier.

(la somme des notes sera calculée au fur et à mesure de la saisie)



University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Exercice 2 : do, switch

Faire un programme simulant une calculatrice 4 opérations (saisir un nombre, une opération, et un nombre, puis afficher le résultat). = pour terminer le programme.

Exemple :

$2 * 5 \rightarrow 10$

$4 = 6 \rightarrow$ fin du programme



University of Geneva
www.miralab.ch

LCI
Introduction au langage C

Exercice 3 : while

Faire un programme qui affiche une somme ajoutant 2 à chaque fois, jusqu'à un résultat maximum donnée par l'utilisateur (la somme sera calculée par additions successives)

Exemple :

pour 7 \rightarrow affichage de 0, 2, 4, 6

pour 4 \rightarrow affichage de 0, 2, 4



University of Geneva
www.miralab.ch

LCI
Introduction au langage C