

# Algo/Prog – DUT 1

Notes

---

---

---

---

---

---

---

---

- 1 Introduction
- 2 Premiers programmes
- 1 Définition de procédures et fonctions
- 2 Conditionnelle (vu en TP)
- 1 Instructions complexe (boucles)
- 1 Variables, types
- 2 Représentation des données en mémoire
- 1 Pointeurs, etc

Notes

---

---

---

---

---

---

---

---

## Organisation du cours

En règle général :

- Découpage : 1H de cours : (Pierre Courtieu)  
2H de TP : (Pierre Courtieu + Rachid Rebiha)
- Petit projet en fin de semestre

N.B. Naturellement la présence est obligatoire à toutes les séances sauf avis médical.

Notes

---

---

---

---




---

---

---

---

## Bibliographie

-  Kernighan et Ritchie (1988, 2<sup>e</sup> édition).  
*The C Programming Language*.  
La bible.  
[fr.wikipedia.org/wiki/The.C.Programming.Language](http://fr.wikipedia.org/wiki/The.C.Programming.Language).
-  Anne Canteaut.  
*La programmation en C*.  
Poly en ligne bien fait (derniers chapitres hors sujet pour nous).  
[www.rocq.inria.fr/secret/Anne.Canteaut/COURS.C/](http://www.rocq.inria.fr/secret/Anne.Canteaut/COURS.C/).
-  *Interactive C tutorial*.  
Cours interactif en anglais.  
[www.learn-c.org/en/](http://www.learn-c.org/en/).

Notes

---

---

---

---

---

---

---

---

## Des programmes partout

Où ?

- ordinateurs (word, excel, call of duty, SAP)
- serveurs (apache, sendmail, etc)
- smartphone, hifi, photo, cafetière, pacemaker, prothèses
- automobile : jusqu'à 70 calculateurs et 1 million de lignes de code
- avions, trains...
- bientôt serrures, radiateurs, ampoules,...

« Le citoyen d'un pays développé utilise quotidiennement et de manière transparente en moyenne 100 processeurs. »

## Qu'est-ce qu'un programme ?

Wikipedia :

« Un programme informatique est un ensemble d'opérations destinées à être exécutées par un ordinateur. »

- Programme source (langage de programmation).  
Compilable (⇒ binaire) (C,fortran) ou interprétable (Java,Python).
- Programme binaire (langage machine).  
exécutable par un microprocesseur.

Le système d'exploitation<sup>1</sup> « démarre » un programme binaire : ordonne au processeur d'exécuter les instructions du programme.

1. (windows, macos, ios, linux, android,...)

## Qu'est-ce qu'un programme (séquentiel) ?

Une recette, comme en cuisine. Une étape après l'autre.

- 1 Pelez et hachez finement 1/4 oignon(s).
- 2 Dans un bol :
  - 1 battez 2 oeufs entiers
  - 2 ajoutez 1 branche de basilic ciselé
  - 3 ajoutez 1/4 cube de bouillon de volaille.
  - 4 ajoutez sel et poivre à votre goût
  - 5 mélangez bien au fouet.
- 3 Dans une poêle :
  - 1 faites chauffer l'huile d'olive
  - 2 faites revenir les oignons hachés pendant 2 min, en remuant
- 4 Versez dans la poêle les oeufs et le fromage râpé laissez cuire 7 min.

## Écrire une recette ≠ faire la cuisine

- Une recette écrite par un auteur :
  - ▶ est un morceau de texte
- Une exécution par un cuisinier :
  - ▶ lit la recette
  - ▶ réalise les actions décrites

Notes

Notes

Notes

Notes

# Écrire une recette ≠ faire la cuisine

- Un programme écrit par un programmeur :
  - ▶ est un morceau de texte
- Une exécution par un processeur :
  - ▶ lit le programme
  - ▶ réalise les actions décrites
- 1 programme, ∞ exécutions différentes
  - ▶ paramètres différents
  - ▶ entrées différents (clavier, souris, fichier, etc)

Notes

---

---

---

---

---

---

---

---

---

---

## 1 programme, ∞ exécutions différentes

Notes

---

---

---

---

---

---

---

---

---

---

Ne pas confondre :

- Le programmeur qui programme :
  - ▶ prévoit chaque cas possibles à l'avance
  - ▶ décrit la (sous-)recette à suivre dans chacun des cas
- Le processeur qui exécute
  - ▶ ne fait que suivre la « recette » pas à pas

## Exemple de programme machine (hexadécimal)

```
00000008 fc 31 c0 8a c0 0e e8 8e d0 bc 00 7c 89 e8 bf 00 |.l.....|
00000010 06 b9 00 01 f3 a5 89 fd b1 08 f3 ab fe 45 f2 e9 |.....E..|
00000020 00 8a fe 46 bb 20 75 08 84 d2 78 07 80 4e bb 40 |...F.u...x.N.0|
00000030 8a 56 ba 88 56 00 a8 fc 00 52 bb c2 07 31 d2 88 |.V.V...R...l..|
00000040 6f fc 0f a3 56 bb 73 19 8a 07 bf 87 07 b1 03 f2 |o...V.s.....|
00000050 ae 74 0e b1 0b f2 ae 83 c7 09 8a 0d 01 cf e8 c5 |.t.....|
00000060 00 42 89 c3 10 73 88 58 26 7f 2a 06 75 04 72 05 |.E...sX...u.r.|
00000070 48 74 0d 30 c0 04 b0 88 46 b8 bf b2 07 e8 a6 00 |H.R...F.....|
00000080 be 7b 07 e8 b2 00 8a 56 b9 4e a8 8e 00 eb 05 b0 |.K...V.N.....|
00000090 07 e8 b0 00 30 e4 cd 1a 89 d7 03 7e bc b4 01 cd |...0.....|
000000a0 16 75 0d 30 e4 cd 1a 39 fa 72 f2 8a 46 b9 eb 16 |.u.0...9.r..F..|
000000b0 30 e4 cd 16 88 e0 2c 1c 74 f1 2c 3c 04 78 06 |0...<.t...<.y..|
000000c0 2c c7 3c 04 77 c9 98 0f a3 46 0c 73 c2 88 46 b9 |.S.W...F$.F..|
000000d0 be 00 08 8a 14 89 f3 9c 04 9c 74 0a c0 e0 04 05 |.....<.t.....|
000000e0 be 07 93 c6 07 80 53 f6 46 bb 40 75 08 bb 00 06 |.....S.F@.....|
000000f0 b4 03 e8 59 00 5e 9d 75 06 8a 56 b8 80 ea 30 bb |...Y.^u..V...0..|
00000100 00 7c b4 02 e8 47 00 72 86 81 bf fe 01 55 aa 0f |...G.r...U..|
00000110 85 7c ff be 85 07 a9 19 00 ff a3 b0 46 e8 24 00 |.....F$.f..|
00000120 b0 31 00 40 eb 17 0f ab 56 0c be 78 07 e8 eb ff |.....V.x...d..|
00000130 89 fe e8 03 00 be 85 07 ac a8 80 75 05 e8 04 00 |.....u.....|
00000140 eb fe 24 7f 53 bb 07 00 b4 0e cd 10 5b c3 8a 74 |.$.S.....[.t..|
00000150 01 8b 4c 02 b0 01 56 89 e7 f6 46 bb 80 74 13 66 |...L.V...F..t.f|
00000160 6a 00 66 ff 74 08 06 53 6a 01 6a 10 89 e6 48 80 |J.f.t..$J...H..|
00000170 cc 40 cd 13 89 fc 5e c3 20 20 a0 0a 44 65 86 61 |e.g...>...Defa|
00000180 75 6c 74 3a a0 0d 8a 00 05 0f 01 06 07 0b 0c 0e |ult:.....|
00000190 83 a5 a6 a9 04 0c 0b 0a 09 08 0a 0e 11 10 01 3f |.....?|
000001a0 bf 44 4f d3 4c 69 e8 75 f8 46 72 65 65 42 53 c4 |.00.Linu.FreeBS..|
000001b0 66 bb 44 72 69 76 e5 20 00 00 80 8f b5 09 00 00 |f.Drive.....|
000001c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
000001f0 00 00 00 00 00 00 00 00 00 00 00 00 55 aa |.....U..|
00000200
```

Notes

---

---

---

---

---

---

---

---

---

---

## Exemple de programme machine (assembleur)

```
.data
UnNom :
.long 43,54,32,76 /* 4 entiers. */ Commentaire (sans effet)
.globl _start
_start: Début des instructions
movl $5, %eax /* EAX nombre d'entiers restant à additionner */
movl $0, %ebx /* EBX va contenir la somme de ces entiers */
movl $UnNom, %ecx /* ECX << pointe >> sur l'élément à additionner */
top:
addl (%ecx), %ebx /* Additionne EBX ECX, résultat dans EBX */
addl $4, %ecx /* Déplace le <<pointeur>> sur le suivant */
decl %eax /* Décréménte le compteur EAX */
jnz top /* Si EAX non nul <<sauter>> à top: */
done:
movl %ebx, UnAutre /* sinon, le resultat est stocké */
movl $0,%ebx /* Ces instructions permettent d'invoquer de */
movl $1,%eax /* terminer l'exécution d'un programme */
int $0x80 /* assembleur et sont indispensable */
```

Notes

---

---

---

---

---

---

---

---

---

---

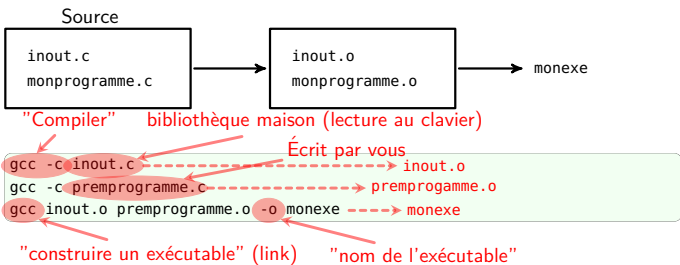
## exemple de programme C

```
#include <stdio.h>
void main(void)
{
    int i;
    char prenom[30];
    FILE *FICHIER1;
    FICHIER1=fopen("/home/chr/premierfichier.txt","w");
    for(i=1;i<=10;i++)
    {
        printf("Rentrez un prénom :\n");scanf("%s",prenom);
        fprintf(FICHIER1,"%s\n",prenom);
    }
    fclose(FICHIER1);
}
```

## Premier programme

```
#include "inout.h" ← Permet écrireString, etc
int main(int nargs, char **args) ← Fonction principale
{ ← Début du programme
    écrireString("Hello world!\n"); ← Première instruction
    écrireString("Bye world!\n");
}
```

## En ligne de commande (unix,gcc)



- répertoire courant : inout.c, inout.h et preprogramme.c
- link : tous les fichiers .o d'un seul coup, ordre important
- raccourci quand un seul fichier : `gcc monpogramme.c -o monexe`
- Exécution : `./monexe`

DÉMO

## Instructions

```
écrireString("Hello world\n");
```

### Une instruction

- morceau de programme/recette = morceau de texte,
- se termine (en C, C++, Java) par un point-virgule : `;`
- opération réalisée à l'exécution,

### Exemples :

- `<< battez 2 oeufs entiers >>`
- `écrireString("Toto\n");`
- `x = y + 3;`
- `return (3);`

### Contre-exemples :

- `<< 2 oeufs >>`
- `"Toto\n"`
- `y+3`
- `(3)`

Notes

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---

## Appel de procédure

```
ecrireString("Hello world\n");  
= appel à la procédure ecrireString
```

- 1 Séparez les feuilles de la salade
- 2 Lavez les feuilles
- 3 **Faites une vinaigrette (p.12)**
- 4 égouttez les feuilles
- 5 ...

## Appel de procédure I

Salade verte :

1. Séparez les feuilles
2. Lavez les feuilles
3. **Faites une vinaigrette**
9. égouttez les feuilles
10. ...

Vinaigrette :

4. versez l'huile dans un bol
5. versez le vinaigre
6. versez la moutarde
7. battez jusqu'à émulsion
8. salez poivrez

- arrête l'exécution de la recette actuelle
- démarre l'autre recette
- autre recette finie ⇒ redémarre l'actuelle

## Deuxième programme (procédure `pause()`)

procédure `pause()` attend une pression sur « entrée ».

```
#include "inout.h"  
  
int main(int nargs, char **args)  
{  
    écrireString("Hello world! Please press enter\n");  
    pause();  
    écrireString("Bye world!\n");  
}
```

## Appel de procédure II – Paramètres

```
ecrireString("Hello world\n");  
ecrireInt(12);  
paramètres
```

Omelette salade

- 1 Séparez les feuilles de la salade
- 2 Lavez les feuilles
- 3 Faites une vinaigrette (p.12)
- 4 **Faites une omelette pour 4 (p.23)**
- 5 égouttez les feuilles
- 6 ...

Notes

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---

## Paramètre de procédures

Omelette pour 1 personne(s)

- 1 Pelez et hachez finement 1/4 oignon(s)
- 2 Dans un bol :
  - 1 battez 2 oeufs entiers
  - 2 ajoutez 1 branche(s) de basilic ciselé
  - 3 ajoutez 1/4 cube(s) de bouillon de volaille.
  - 4 ajoutez sel et poivre à votre goût
  - 5 mélangez bien au fouet.
- 3 Dans une poêle :
  - 1 faites chauffer l'huile d'olive
  - 2 faites revenir les oignons hachés pendant 2 min, en remuant
- 4 Versez dans la poêle les oeufs et le fromage râpé laissez cuire 7 min.

Notes

---

---

---

---

---

---

---

---

## Paramètre de procédures

⇒ Paramètre  $n$  (formel)

Omelette pour  $n$  personne(s)

- 1 Pelez et hachez finement  $n/4$  oignon(s)
- 2 Dans un bol :
  - 1 battez  $2 \times n$  oeufs entiers
  - 2 ajoutez  $n$  branche(s) de basilic ciselé
  - 3 ajoutez  $n/4$  cube(s) de bouillon de volaille.
  - 4 ajoutez sel et poivre à votre goût
  - 5 mélangez bien au fouet.
- 3 Dans une poêle :
  - 1 faites chauffer l'huile d'olive
  - 2 faites revenir les oignons hachés pendant 2 min, en remuant
- 4 Versez dans la poêle les oeufs et le fromage râpé laissez cuire 7 min.

Notes

---

---

---

---

---

---

---

---

## Paramètre de procédures

Omelette pour 3 ← paramètre réel

Omelette pour 3 personne(s)

- 1 Pelez et hachez finement 3/4 oignon(s)
- 2 Dans un bol :
  - 1 battez  $2 \times 3$  oeufs entiers
  - 2 ajoutez 3 branche(s) de basilic ciselé
  - 3 ajoutez 3/4 cube(s) de bouillon de volaille.
  - 4 ajoutez sel et poivre à votre goût
  - 5 mélangez bien au fouet.
- 3 Dans une poêle :
  - 1 faites chauffer l'huile d'olive
  - 2 faites revenir les oignons hachés pendant 2 min, en remuant
- 4 Versez dans la poêle les oeufs et le fromage râpé laissez cuire 7 min.

Notes

---

---

---

---

---

---

---

---

## Paramètre de procédures

Omelette pour 4

Omelette pour 4 personne(s)

- 1 Pelez et hachez finement 4/4 oignon(s)
- 2 Dans un bol :
  - 1 battez  $2 \times 4$  oeufs entiers
  - 2 ajoutez 4 branche(s) de basilic ciselé
  - 3 ajoutez 4/4 cube(s) de bouillon de volaille.
  - 4 ajoutez sel et poivre à votre goût
  - 5 mélangez bien au fouet.
- 3 Dans une poêle :
  - 1 faites chauffer l'huile d'olive
  - 2 faites revenir les oignons hachés pendant 2 min, en remuant
- 4 Versez dans la poêle les oeufs et le fromage râpé laissez cuire 7 min.

Notes

---

---

---

---

---

---

---

---

## Troisième programme

```
#include "inout.h"

int main(int nargs, char **args) {
    ecrireString("\nHello world!\n");
    pause();
    ecrireInt(12); // Écrit un entier à l'écran
    ecrireString("\nBye world!\n");
}
```

Notes

---

---

---

---

---

---

---

---

## Quatrième programme

```
#include "inout.h"

int main(int nargs, char **args) {
    ecrireString("\nHello world!\n");
    pause();
    ecrireInt(12+5); // entier
    ecrireString("\nBye world!\n");
}
```

Notes

---

---

---

---

---

---

---

---

## Appel de procédure I

Salade verte pour 3 :

1. Séparez les feuilles de la salade
2. Lavez les feuilles
3. **Faites une vinaigrette pour 3**
9. égouttez les feuilles
10. ...

Vinaigrette pour **n3** :

4. versez 3 cuillère d'huile dans un bol
5. versez 3/3 cuillères de vinaigre
6. versez 3/3 cuillères de moutarde
7. battre jusqu'à émulsion
8. salez poivrez

- arrête l'exécution de la recette actuelle
- démarre l'autre recette **avec le paramètre réel**
- autre recette finie ⇒ redémarre l'actuelle

Notes

---

---

---

---

---

---

---

---

## fonctions et variables

`lireInt()` : lit l'entier tapé par l'utilisateur

- Attend que l'utilisateur tape un entier (ex : 123) + « entrée »
- « retourne » l'entier tapé
- `lireInt()` **équivalent à un entier**
- `ecrireInt(lireInt())` (démonstration)
- `ecrireInt(lireInt() + lireInt())` (démonstration)

Notes

---

---

---

---

---

---

---

---

## Appel de fonction

Salade verte pour 3 :

Vinaigrette pour 3 :

<ol style="list-style-type: none"><li>Séparez les feuilles de la salade</li><li>Lavez les feuilles</li><li>égouttez les feuilles</li><li>Faites une vinaigrette pour 3 et versez la vinaigr.</li><li>...</li></ol>	<ol style="list-style-type: none"><li>versez 3 cuillère d'huile dans un bol</li><li>versez 3/3 cuillères de vinaigre</li><li>versez 3/3 cuillères de moutarde</li><li>battre jusqu'à émulsion</li><li>salez poivrez</li></ol>
--	---

- arrête l'exécution de la recette actuelle
- instruction 4. pas finie!
- démarre l'autre recette avec le paramètre réel
- autre recette finie ⇒ instruction 4. redémarre en utilisant la valeur retournée

Notes

---

---

---

---

---

---

---

---

## Variables

```
lireInt();
```

- « retourne » l'entier tapé
- `ecrireInt(lireInt()+lireInt())` fait 2 lectures clavier.
- comment faire pour utiliser plusieurs fois le résultat du premier `lireInt()` ?
- Exemple : lire un entier  $n$  au clavier, puis afficher la valeur de  $n + n^2$ .
- Stockage du résultat de (= valeur retournée par) `lireInt()` pour l'utiliser plus tard.

Notes

---

---

---

---

---

---

---

---

## Variable

- Bout de mémoire : VARIABLE
- Suffisamment grande pour contenir un `int`
- Déclaration : `int n;`
- Affectation : `n = 12;`, `n=3+4;`, `n=lireInt()+7;`...
  - ▶ signe « égal » (=) ne signifie pas égalité
  - ▶ `x = 12;` signifie « Mettre la valeur 12 dans la variable x ».
- Utilisation : `n` (ex : `n + n*n`)

Notes

---

---

---

---

---

---

---

---

## Variable

L'instruction d'affectation :

```
x = f(12) + 7 - g(4)
```

S'exécute de la manière suivante :

- ➊ Évaluation à droite du « = » : `f(12) + 7 - g(4)`
- ➋ Valeur obtenue stockée dans `x` (valeur précédente écrasée).

Question : Que fait l'instruction :

```
x = x + 2;
```

Notes

---

---

---

---

---

---

---

---