

Conditionnelles (`if else switch`) et logique booléenne en C

Séance 3

de l'UE « introduction à la programmation »

Bruno Bouzy

bruno.bouzy@parisdescartes.fr

if

```
int x;  
printf("Tapez une valeur: ");  
scanf("%d", &x);  
  
if (x>0)  
    printf("valeur > 0.\n");  
else  
    printf("valeur <= 0.\n");
```

conditions (if, else, switch) et
logique booléenne

if

- 1ère exécution: ProgC > ./a.out
Tapez une valeur: **6**
Valeur > 0.
- 2ème exécution: ProgC > ./a.out
Tapez une valeur: **-6**
Valeur <= 0.
- 3ème exécution: ProgC > ./a.out
Tapez une valeur: **0**
Valeur <= 0.

conditions (if, else, switch) et
logique booléenne

if

- Bloc d'instructions après le `if` et/ou le `else`:

```
if (x>0) {  
    x = 3*x;  
    printf("x = %d\n", x);  
}  
else {  
    x = 2*x-1;  
    printf("x = %d\n", x);  
}
```

conditions (if, else, switch) et
logique booléenne

if

- Imbrications de `if` et/ou `else`:

```
if (x>0) {  
    x = 3*x;  
    if (y>x) y = x;  
}  
else {  
    x = 2*x-1;  
    if (y>x) y = x;  
    else y = 0;  
}
```

conditions (if, else, switch) et
logique booléenne

if

- else if :

```
if (x>0) {  
    x = 3*x;  
    y = x;  
}  
else if (x<0) {  
    x = 2*x-1;  
    y = 0;  
}  
else y = 0;
```

conditions (if, else, switch) et
logique booléenne

switch

```
int x;
printf("Tapez une valeur: ");
scanf("%d", &x);

switch(x) {
case -1: printf("valeur -1.\n");
    break;
case 0:  printf("valeur nulle.\n");
    break;
case 1:  printf("valeur +1.\n");
    break;
default: printf("valeur ni -1, ni 0,
                    ni +1.\n");
    break;
}
```

conditions (if, else, switch) et
logique booléenne

switch

- 1ère exécution: Tapez une valeur: **+1**
valeur +1.
- 2ème exécution: Tapez une valeur: **-1**
valeur -1.
- 3ème exécution: Tapez une valeur: **0**
valeur nulle.
- 4ème exécution: Tapez une valeur: **7**
valeur ni -1, ni 0,
ni +1.

conditions (if, else, switch) et
logique booléenne

Affectation conditionnelle

```
int valSiOui;
printf("valeur si oui ? ");
scanf("%d", &valSiOui);
int valSiNon;
printf("valeur si non ? ");
scanf("%d", &valSiNon);
int c;
printf("Oui (1) ou non (0) ? ");
scanf("%d", &c);
int v = (c==1) ? valSiOui : valSiNon;
printf("v = %d\n", v);
```

conditions (if, else, switch) et
logique booléenne

Affectation conditionnelle

- 1ère exécution: valeur si oui ? **6**
valeur si non ? **-5**
Oui (1) ou non (0) ? **1**
v = **6**
- 2ème exécution: valeur si oui ? **6**
valeur si non ? **-5**
Oui (1) ou non (0) ? **0**
v = **-5**

conditions (if, else, switch) et
logique booléenne

Variables booléennes

- Mauvaise nouvelle: pas de type booléen prédéfini en C
- On le simule avec le type entier
- Convention:
 - 0 signifie « faux »
 - 1 ou toute valeur différente de 0 signifie « vrai »

conditions (if, else, switch) et
logique booléenne

Exemple 1 (1/2)

- `if, else, vrai et faux`

```
int x=1, y=0;
```

```
printf("x = %d\n", x);  
printf("y = %d\n", y);
```

```
if (x) printf("x est vrai.\n");  
else   printf("x est faux.\n");
```

```
if (y) printf("y est vrai.\n");  
else   printf("y est faux.\n");
```

conditions (if, else, switch) et
logique booléenne

Exemple 1 (2/2)

- Sortie écran:

x = 1

y = 0

x est **vrai**.

y est **faux**.

conditions (if, else, switch) et
logique booléenne

Exemple 2 (1/2)

- **if, else, vrai et faux (suite...)**

```
int x=1, y=0;
printf("x = %d, y = %d\n", x, y);

if (x) {
    printf("x est vrai.\n");
    if (y) printf("y est vrai.\n");
    else   printf("y est faux.\n");
}
else {
    printf("x est faux.\n");
    if (y) printf("y est vrai.\n");
    else   printf("y est faux.\n");
}
```

conditions (if, else, switch) et
logique booléenne

Exemple 2 (2/2)

- Sortie écran:

```
x = 1
y = 0
x est vrai.
y est faux.
```

conditions (if, else, switch) et
logique booléenne

Opérateurs booléens (1/2)

- `!`, `&&`, `||` : non logique, et logique, ou logique

```
char a=0, b=1, c=0, d=1;
```

```
printf("a = %d, b = %d, c = %d, d = %d.\n", a, b, c, d);
```

```
printf("a ET b = %d, ", a && b);
```

```
printf("a ET c = %d, ", a && c);
```

```
printf("b ET d = %d. \n", b && d);
```

```
printf("a OU b = %d, ", a || b);
```

```
printf("a OU c = %d, ", a || c);
```

```
printf("b OU d = %d. \n", b || d);
```

```
printf("NON a = %d, ", !a);
```

```
printf("NON b = %d. \n", !b);
```

conditions (if, else, switch) et
logique booléenne

Opérateurs booléens (2/2)

- Sortie écran:

`a = 0, b = 1, c = 0, d = 1.`

`a ET b = 0, a ET c = 0, b ET d = 1.`

`a OU b = 1, a OU c = 0, b OU d = 1.`

`NON a = 1, NON b = 0.`

conditions (if, else, switch) et
logique booléenne

Opérateurs de comparaison (1/4)

- `if, else, test de comparaison ==, >=, <=`

```
int x=4, y=5, z=7;
```

```
printf("x = %d, y = %d, z = %d\n", x, y, z);
```

```
if (x==4) printf("x vaut 4.\n");  
else      printf("x ne vaut pas 4.\n");
```

```
if (y>=4) printf("y est superieur ou egal a 4.\n");  
else      printf("y n'est pas superieur ou egal a 4.\n");
```

```
if (z<=4) printf("z est inferieur ou egal a 4.\n");  
else      printf("z n'est pas inferieur ou egal a 4.\n");
```

conditions (if, else, switch) et
logique booléenne

Opérateurs de comparaison (2/4)

- Sortie écran:

```
x = 4, y = 5, z = 7
```

```
x vaut 4.
```

```
y est supérieur ou égal a 4.
```

```
z n'est pas inférieur ou égal a 4.
```

Opérateurs de comparaison (3/4)

- Donner la sortie de:

```
printf ("4-5-3==2*-5/2+1 = %d\n",  
        4-5-3==2*-5/2+1);
```

conditions (if, else, switch) et
logique booléenne

Opérateurs de comparaison (4/4)

- Sortie:

$$4 - 5 - 3 == 2 * -5 / 2 + 1 = \mathbf{1}$$

Opérateurs booléens et comparaison (1/4)

- `if, else, ==, >=, <=, &&, ||`

```
int x=4, y=5;
```

```
printf("x = %d, y = %d\n", x, y);
```

```
if ((x==4) && (y>=4))
```

```
    printf("x vaut 4 et y est supérieur ou égal à 4.\n");
```

```
else
```

```
    printf("x ne vaut pas 4 ou y n'est pas supérieur ou égal à 4.\n");
```

```
if ((x==4) || (y>=4))
```

```
    printf("x vaut 4 ou y est supérieur ou égal à 4.\n");
```

```
else
```

```
    printf("x ne vaut pas 4 et y n'est pas supérieur ou égal à 4.\n");
```

conditions (if, else, switch) et
logique booléenne

Opérateurs booléens et comparaison (2/4)

- Sortie écran:

x = 4, y = 5

x vaut 4 **et** y est supérieur ou égal à 4.

x vaut 4 **ou** y est supérieur ou égal à 4.

Opérateurs booléens et comparaison (3/4)

`if, else, ==, >=, <=, &&, || (bis)`

```
x = 4; y = 3;
```

```
printf("x = %d, y = %d\n", x, y);
```

```
if ((x==4) && (y>=4))
```

```
    printf("x vaut 4 et y est supérieur ou égal à 4.\n");
```

```
else
```

```
    printf("x ne vaut pas 4 ou y n'est pas supérieur ou égal à 4.\n");
```

```
if ((x==4) || (y>=4))
```

```
    printf("x vaut 4 ou y est supérieur ou égal à 4.\n");
```

```
else
```

```
    printf("x ne vaut pas 4 et y n'est pas supérieur ou égal à 4.\n");
```

conditions (if, else, switch) et
logique booléenne

Opérateurs booléens et comparaison (4/4)

- Sortie écran:

`x = 4, y = 3`

`x ne vaut pas 4 ou y n'est pas supérieur ou égal a 4.`

`x vaut 4 ou y est supérieur ou égal a 4.`

biblio.c

- Ecrire un programme permettant à l'utilisateur de:
 - Entrer un nombre total de livres (`ntl`),
 - Entrer un nombre de livres empruntés (`nle`),
 - Voir le pourcentage de livres restants.
- Cas d'erreurs:
 - Si `ntl` est négatif ou nul, le programme s'arrête et affiche une erreur.
 - Si `nle` est négatif, le programme s'arrête et affiche une erreur.
 - Si `nle` est supérieur à `ntl`, le programme s'arrête et affiche une erreur.

conditions (if, else, switch) et
logique booléenne

biblio.c

- **Cas 1:**

```
n total livres ? -100
```

```
erreur: n total < 0
```

- **Cas 2:**

```
n total livres ? 600
```

```
n livres empruntes ? 900
```

```
erreur: n emprunts > n total
```

- **Cas normal:**

```
n total livres ? 600
```

```
n livres empruntes ? 111
```

```
pc livres restants = 81.50
```

conditions (if, else, switch) et
logique booléenne

biblio.c

```
#include <stdio.h>
int main() {
    float ntl, nle;
    printf("n total livres ? ");
    scanf("%f", &ntl);
    if (ntl<0)
        printf("erreur: n total < 0\n");
    else {
        printf("n livres empruntes ? ");
        scanf("%f", &nle);
        if (nle>ntl)
            printf("erreur: n emprunts > n total\n");
        else if (nle<0)
            printf("erreur: n emprunts < 0\n");
        else
            printf("pc l restants = %.2f\n", 100*(ntl-nle)/ntl);
    }
    return (0);
}
```

conditions (if, else, switch) et
logique booléenne

biblio.c

```
#include <stdio.h>
int main() {
float ntl, nle;
printf("n total livres ? ");
scanf("%f", &ntl);
if (ntl<0) printf("erreur: n total < 0\n"); else {
printf("n livres empruntes ? ");
scanf("%f", &nle);
if (nle>ntl) printf("erreur: n emprunts > n total\n");
else if (nle<0) printf("erreur: n emprunts < 0\n");
else printf("pc l restants = %.2f\n", 100*(ntl-nle)/ntl); }
return (0);
}
```

- **Compilable ? Lisible ?**

conditions (if, else, switch) et
logique booléenne

Dernier (mauvais?) exemple (1/2)

- Donner la sortie de:

```
printf("32&&2.3 = %d\n", 32&&2.3);
```

```
printf("!65.34 = %d\n", !65.34);
```

```
printf("0||!(32>12) = %d\n",  
       0||!(32>12));
```

Dernier (mauvais?) exemple (2/2)

- Sortie:

$$32 \& \& 2.3 = \mathbf{1}$$

$$!65.34 = \mathbf{0}$$

$$0 || !(32 > 12) = \mathbf{0}$$

conditions (if, else, switch) et
logique booléenne

Résumé de la séance 3

- Conditionnelles

`if, else, switch`

- Affectation conditionnelle

`x = c ? y : z;`

- Variables « booléennes »

- `if, else` avec opérateurs

- booléens

- de comparaison

conditions (if, else, switch) et
logique booléenne