

Informatique S1
Programmation C

- *Objectifs* : réaliser des tests simples et emboîtés
- Présentation de l'instruction `if` :
 - `if` et `if ... else`
- Usage de tests emboîtés
- Opérateurs logiques
 - `&&`, `||`

Instruction `if`

- Instruction de contrôle permettant de faire un test
- Format :


```
if (test)
{
    Bloc d'instructions ;
}
```

Le bloc d'instructions n'est exécuté que si la condition se vérifie **VRAI**

Exemple

```

1  #include <stdio.h>
2
3  /* CM 6 : instructions de controle if */
4
5  int main () {
6      int k = 0;
7
8      printf ("Entrez k : ");
9      scanf ("%d", &k);
10
11     if (k < 0) {
12         printf ("k est negatif (%d)\n", k);
13     }
14 }

```

Le `printf` ne sera exécuté que si `k < 0`

Instruction `if ... else`

- Format :


```
if (test)
{
    Bloc d'instructions 1;
}
else
{
    Bloc d'instructions 2;
}
```

Exécuté si le test est **VRAI**

Exécuté si le test est **FAUX**

Exemple

```

1 #include <stdio.h>
2
3 /* CM 6 : instructions de controle if */
4
5 int main () {
6     int k = 0;
7
8     printf ("Entrez k : ");
9     scanf ("%d", &k);
10
11     if (k < 0) {
12         printf ("k est negatif (%d)\n", k);
13     }
14     else {
15         printf ("k est positif (%d)\n", k);
16     }
17 }
18
19

```

Si $k < 0$, le `printf` à la ligne 12 s'affiche (" k est négatif ")

Sinon, le `printf` à la ligne 15 s'affiche (" k est positif ")

Instructions *if* emboîtées

- Une instruction `if` peut contenir d'autres instructions de contrôle
– *if, while, do...while, for...*

```

if (test 1) {
    if (test 2)
        { instructions if 2; }
    else
        { instructions else if 2; }
    ...
}
else {
    instructions else if 1;
    ...
}

```

```

if (test 1) {
    instructions if 1;
}
else {
    if (test 2)
        { instructions if 2; }
    else
        { instructions else if 2; }
    ...
}

```

Exemple

```

1 #include <stdio.h>
2
3 /* CM 6 : instructions de controle if */
4
5 int main () {
6     int k = 0;
7
8     printf ("Entrez k : ");
9     scanf ("%d", &k);
10
11     if (k < 0) {
12         printf ("k est negatif (%d)\n", k);
13     }
14     else {
15         printf ("k est positif (%d)\n", k);
16
17         if (k < 5)
18             printf ("\t k est inferieur a 5 \n");
19         else
20             printf ("\t k est superieur ou egal a 5 \n");
21     }
22 }

```

On n'atteint le 2° `if` que si $k < 0$ est faux

Opérateur logiques

- Opérateurs de la logique booléenne :
 - Opérateur `&&` \Rightarrow ET (AND)
 $(a == 0) \ \&\& \ (a < 5)$
 - Opérateur `||` \Rightarrow OU (OR)
 $(c == 'o') \ || \ (c == 'n')$
 - Opérateur `!` \Rightarrow NON (NOT)
 $!(c == 'o')$

Opérateurs logiques

&&	Vrai	Faux
Vrai	V	F
Faux	F	F

a = 0, b = 1, c = -1
(a < b) && (b < c) ?

	Vrai	Faux
Vrai	V	V
Faux	V	F

(a < b) || (b < c) ?

!	Vrai	Faux
	F	V

!(a < b) ?

Exemple

```

1  #include <stdio.h>
2
3  /* CM 6 : instructions de controle if */
4
5  int main () {
6      int k = 0;
7      printf ("Entrez k : ");
8      scanf ("%d", &k);
9
10
11     if ( (k >= 0) && (k <= 5) ) {
12         printf ("k est positif et inferieur (ou egal) a 5 (%d)\n", k);
13     }
14     else if ( (k > 5) ) {
15         printf ("k est positif et superieur a 5 (%d)\n", k);
16     }
17     else {
18         printf ("k est negatif (%d)\n", k);
19     }
20 }

```

(k >= 0) && (k <= 5)
Valeur de k entre 0 et 5