

* Réponse sur viz2.

* Ajouter des exos supplémentaires

* Titulaire en informatique.

* Options: overissements
additionnels.

$x = y;$ // affectation.

$x == y$ // Comparaison
d'égalité

$x > y$

$x + 2$

$x > y \Rightarrow \begin{matrix} \text{vrai} \\ \text{faux} \end{matrix} \text{ } \text{booléens.}$

\emptyset : faux
n'est pas \emptyset , vrai

```
int x;
```

```
x = 25; //
```

```
if (x)
```

```
{
```

```
    printf("Bonjour");
```

```
}
```

,

$$\underline{x > 0} \Rightarrow \emptyset \text{ si } x_{pas} > 0$$

$$\Rightarrow \underline{1}$$

$$=$$

if ($x > 0$)

{

≡

}

$x > 0$

||

\emptyset

int

si c'est

so even
false

valid

if ($x > 0$)

if $x \geq 25$

①

if $x \geq -2$

②

}

}

if ($x \geq y$)

~~$x \geq y$~~

✓ ✓

if ($x \leq y$)

~~$x \leq y$~~

int x = 19;

int y = 17;

int z = 15;

if ($x > y > z$)

{

printf ("x est le
plus grand!");

}

$$\begin{array}{c} 5 + 3 + 9 \\ \hline 8 + 9 \\ \hline 17 \end{array}$$

$$\begin{array}{l} x: 19 \\ y: 17 \\ z: 15 \end{array}$$

$$\begin{array}{c} x > y > z \\ \hline 1 > 2 \\ \hline \emptyset \end{array}$$

$$x > y > z$$

$$[x > y \quad \underline{\underline{\text{ET}}} \quad y > z]$$

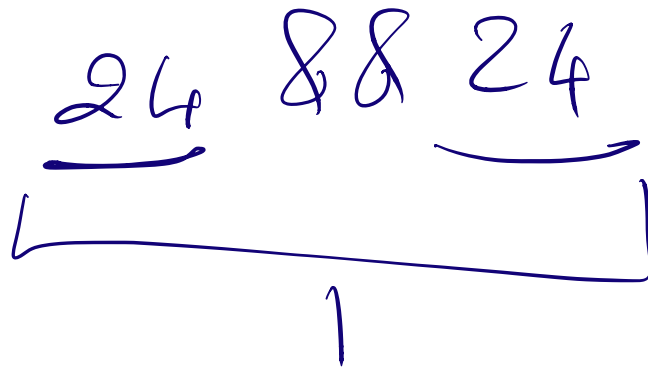
$$\{ \underbrace{x > y}_1 \quad \&\& \quad \underbrace{y > z}_2 \}$$

		cond1 && cond2	
cond1	cond2		
1	1	1	1
1	0	0	0
0	1	0	0
0	0	0	0

Table de
vérité du
&& (1,1)

ϕ , ϕ , ϕ

and 15)



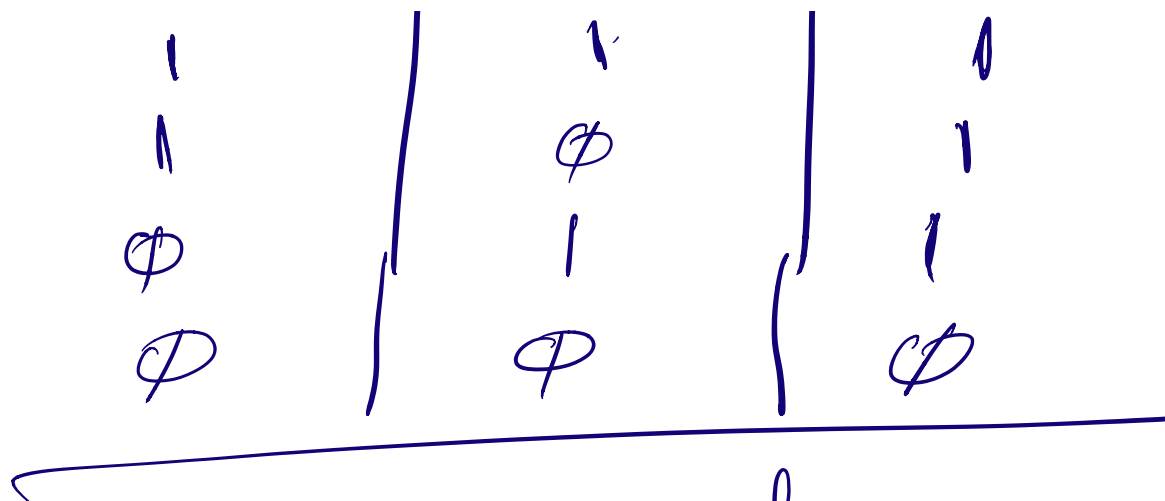
$x = -5$; $y = 19$;

if ($x < 0$ || $y == 7$)
{

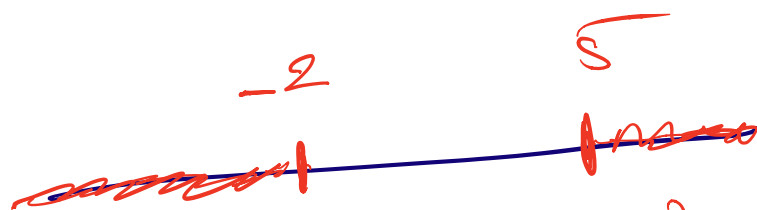
==

}

cond 1		cond 2		cond 1 cond 2
--------	--	--------	--	------------------



saisir une valeur
(x)



$is (x < -2 \parallel x > 5)$

Traitement exécuté si
 x n'est pas supérieur
à 5.

if ($!(x > 5)$)

}
 }
 }

$x = 5;$

$\{ \text{if } (x \neq 0) \rightarrow \}$

}

}

int y;

$x = 5$

,

$y = x; // y = 0;$

cond	!cond
1	ϕ
ϕ	1

! (x)

-x
↑

88

23: 0001011

23 & 32

23: 00001 0111
32: 0010 0000

23 & 32: 0000 0000

23: 00001 0111
24: 00001 0000

00001 0000

23 & 24

$$23 \& 24 = 16$$

$$x = 23;$$

$$y = 24;$$

$$z = x \& y;$$

dd: 0 ou 1

d: entier

if (x > 0 & y < 5)

ou bit à bit :

23 0001 0111

24 0001 1000

23 | 24 0001 | 1111

23 | 24 = 31

ou bit à bit : ~

23: 0001 0111

~23: 111 01000

~23 = 230

x : 1 0 1 0 1 1 1 0
 y : 0 0 0 0 1 0 0 0

~ : Operateur NOT
bit à bit.

$18 =$

0	0	0	1	0	0	1	0
:	:	:	:	:	:	:	:
1	1	1	0	1	1	0	1

$$\sim 18 = 237$$

xor

x	y	xor
1	1	0
0	1	1
1	0	1
0	0	0