Série 0x06

Opérateurs et embranchements

INFO1-TIN 3 mars 2026, 7efbe5

Exercice 1 : Opérateurs combinés

En reprenant à chaque fois les valeurs suivantes, calculer les valeurs de `i`, `j` et lorsque cela s'applique `z` après l'exécution des instructions suivantes :

```
int i = 1, j = 3;
int z;
```

Expression	i	j	z
`i += j`	.....	.....	.....
`i += -j`	.....	.....	.....
`i -= j`	.....	.....	.....
`i -= -j`	.....	.....	.....
`i *= j`	.....	.....	.....
`i *= -j`	.....	.....	.....
`i /= j`	.....	.....	.....
`z = i * j == 6`	.....	.....	.....
`z = i++ * j == 6`	.....	.....	.....
`z = ++i * j == 6`	.....	.....	.....

Exercice 2 : Opérateur ternaire

(a) Simplifiez l'expression suivante.

```
z = (a > b ? a : b) + (a <= b ? a : b) ;
```

(b) Soit variable `n` est de type `int`. Écrire une expression unique qui prend la valeur :

1. `-1` si `n` est négatif
2. `0` si `n` est nul
3. `1` si `n` est positif

Exercice 3 : Opérateurs incorrects

Soit les déclarations suivantes, indiquez pourquoi les propositions suivantes sont incorrectes :

```
double f, g = 7;
int i, j = j;
```

(a)
```
int(f) + 1.9
```

(b)
```
i = 1 + j = j / 2
```

(c)
```
f = g << 2
```

(d)
```
i = ++j++
```

(e)
```
i++ = ++j
```

Exercice 4

Indiquez pour chaque groupe d'instruction ci-dessous si l'expression est correcte ou non. Sinon, expliquer pourquoi.

```
int i;
assert(scanf("%d", &i) == 1);
```

(a)
```
if (!(i < 8) && !(i > 8)) then printf("i vaut 8\n");
```

(b)
```
if (!(i < 8) && !(i > 8)) printf("i "); printf("vaut 8\n");
```

(c)
```
if !(i < 8) && !(i > 8) printf("i vaut 8\n");
```

(d)
```
if (!(i < 8) && !(i > 8)) printf("i vaut 8\n");
```

(e)
```
if (i = 8) printf("i vaut 8\n");
```

(f)
```
if (i & (1 << 3)) printf("i vaut 8\n");
```

(g)
```
if (i ^ 8) printf("i vaut 8\n");
```

(h)
```
if (i - 8) printf("i vaut 8\n");
```

(i)
```
if (i == 1 << 3) printf ("i vaut 8\n");
```

(j)
```
if (!((i < 8) || (i > 8))) printf("i vaut 8\n");
```

Exercice 5 : Analyse de code

Que voyez-vous sur la sortie standard ?

Notez que selon le standard ISO8899, une expression comportant plusieurs post ou pré incrémentation est indéterminée, néanmoins la logique de l'expression est définie dans la plupart des compilateurs et elle suit la règle enseignée en cours.

```
#include <stdio.h>
int main() {
    int x = 2;
    int y = ++x * ++x;
    printf("%d%d", x, y);
    x = 2;
    y = x++ * ++x;
    printf("%d%d", x, y);
}
```