



TD 01 – Variables, expressions, conditions et boucles

Exercice 1 Exécution mentale de code simple

 Pour chacun des extraits de code suivants, déterminez les valeurs finales des variables.

 Attention, il y a des pièges !

1.1

```
int a = 10;
int b = 5;
int c;
double d;
c = b / a;
d = b / a;
```

Variable	Valeur finale
a	
b	
c	
d	

1.2

```
int a = 40;
int b;
int c;
b = a++ + 2;
c = ++a;
```

Variable	Valeur finale
a	
b	
c	

1.3

```
int a = 3;
int b = 8;
bool c;
bool d;
c = a * b == 42;
d = !(a / b);
```

Variable	Valeur finale
a	
b	
c	
d	

1.4

```
int a = 3;
int b = 8;
bool c;
bool d;
bool e;
c = a == 0 && a * b * 2 == 48;
d = a == 0 || a * b * 2 == 48;
e = d && !c || !d && c;
```

Variable	Valeur finale
a	
b	
c	
d	
e	

1.5

```
int a = 0;
int b = 1;
if (a == 0)
    b += 9;
```

Variable	Valeur finale
a	
b	

1.6

```
int a = 0;
int b = 1;
if (a == 1)
    b += 9;
```

Variable	Valeur finale
a	
b	

1.7

```
int a = 0;
int b = 1;
if (a == 0)
    b += 9;
```

Variable	Valeur finale
a	
b	

1.8

```
int a = 0;
int b = 1;
if (a != 0)
    b += 9;
```

Variable	Valeur finale
a	
b	

1.9

```
int a = 0;
int b = 1;
if (a != 0);
    b += 9;
```

Variable	Valeur finale
a	
b	

1.10

```
int a = 0;
int b = 1;
if (a != 0)
    b += 9;
a = 42;
```

Variable	Valeur finale
a	
b	

1.11

```
int a = 0;
int b = 1;
if (a != 0)
{
    b += 9;
    a = 42;
}
```

Variable	Valeur finale
a	
b	

1.12

```
int a = 0;
int b = 1;
if (a++ == 0)
    b = 8;
else
    b = 9;
```

Variable	Valeur finale
a	
b	

1.13

```
int a = 0;
int b = 1;
if (++a == 0)
    b = 8;
else
    b = 9;
```

Variable	Valeur finale
a	
b	

1.14

```
int a = 0;
int b = 1;
if (++a == 0)
{
    b = 8;
}
else
    b = 9;
```

Variable	Valeur finale
a	
b	

1.15

```
int a = 0;
int b = 1;
if (++a == 0)
{
    b = 8;
}
else
{
    b = 9;
}
```

Variable	Valeur finale
a	
b	

1.16

```
int a = 2;
int b = 8;
switch (a)
{
    case 1:
        b *= 1;
    case 2:
        b *= 2;
    case 3:
        b *= 8;
    case 4:
        b *= 16;
    default:
        b = 42;
}
```

Variable	Valeur finale
a	
b	

1.17

```
int a = 2;
int b = 8;
switch (a)
{
    case 1:
        b *= 1;
    case 2:
        b *= 2;
    case 3:
        b *= 8;
        break;
    case 4:
        b *= 16;
    default:
        b = 42;
}
```

Variable	Valeur finale
a	
b	

1.18

```
int a = 2;
int b = 8;
switch (a)
{
case 1:
    b *= 1;
    break;
case 2:
    b *= 2;
    break;
case 3:
    b *= 8;
    break;
case 4:
    b *= 16;
    break;
default:
    b = 42;
}
```

Variable	Valeur finale
a	
b	

1.19

```
int a = 2;
int b = 8;
switch (a)
{
case 1:
    b *= 1;
    break;
case 2:
    int c = 8;
    b *= 2;
    break;
default:
    b = 42;
}
```

Variable	Valeur finale
a	
b	
c	

1.20

```
int a = 2;
int b = 8;
switch (a)
{
case 1:
    b *= 1;
    break;
case 2:
    {
        int c = 8;
        b *= 2;
    }
    break;
default:
    b = 42;
}
```


Variable	Valeur finale
a	
b	
c	

1.21

```
int a = 2;
int b = 8;
switch (a)
{
case 1:
    b *= 1;
    break;
case 2:
    {
        int c = 8;
        b *= 2;
        break;
    }
default:
    b = 42;
}
```

Variable	Valeur finale
a	
b	
c	

Exercice 2 Boucles

 Pour chacun des extraits de code suivants, déterminez ce qui est affiché dans la console :

2.1

```
int a = 5058;
while (a > 0)
{
    std::cout << a % 2;
    a /= 2;
}
```

L'opérateur % calcule le reste de la division de a par 2.

2.2

```
int a = 5058;
int cpt = 0;
while (a > 0)
{
    std::cout << a % 2;
    a /= 2;
    if (++cpt % 4 == 0)
        std::cout << " ";
}
while (cpt++ % 4 != 0)
    std::cout << "0";
```

2.3

```
int a = 5058;
int cpt = 0;
while (a > 0)
{
    int r = a % 16;
    if (r < 10)
        std::cout << r;
    else
    {
        char c = 'A' + r - 10;
        std::cout << c;
    }
    a /= 16;
    if (++cpt % 4 == 0)
        std::cout << " ";
}
while (cpt++ % 4 != 0)
    std::cout << "0";
```

2.4


```
for (int i = 0; i < 10; ++i)
{
    std::cout << i << " ";
}
```


2.5


```


uint32_t a = 2882400001;
for (int i = sizeof(a) * 8 - 1; i >= 0; --i)
{
    uint32_t puissance_2 = 1 << i;
    if ((a & puissance_2) == 0)
        std::cout << '0';
    else
        std::cout << '1';
    if (i % 4 == 0)
        std::cout << ' ';
}

```

 Le type `uint32_t` est un entier non signé sur 32 bits.

 `sizeof(a)` retourne la taille en octets de la variable `a`.

 `a << b` réalise un décalage vers la gauche de `b` bits de `a`. Ex : si `b` vaut 3 (0011 en binaire), alors `b << 2` vaut 12 (1100 en binaire).

 `a & b` réalise l'opération logique ET entre chaque bit de `a` et chaque bit de `b`. Ex 14 & 3 vaut 2 car en binaire 14 s'écrit 1110 et 3 s'écrit 0011 :

$$\begin{array}{r}
 1110 \\
 \& 0011 \\
 \hline
 0010
 \end{array}$$

2.6

```

uint32_t a = 2882400001;
for (int i = sizeof(a) * 2 - 1; i >= 0; --i)
{
    uint32_t masque_chiffre_hexa = 0xF << i * 4;
    uint32_t chiffre_hexa = (masque_chiffre_hexa & a) >> i * 4;
    if (chiffre_hexa < 10)
        std::cout << chiffre_hexa;
    else
    {
        char c = 'A' + chiffre_hexa - 10;
        std::cout << c;
    }
    if (i % 2 == 0)
        std::cout << ' ';
}

```


2.7

```
int i, j;
for (i = 1; i <= 3; ++i)
{
    for (j = 1; j <= 3; ++j)
    {
        if (i < j)
            std::cout << i;
        else
            std::cout << j;
    }
}
```

2.8

```
for (int i = 0; i < 5; i++)
{
    for (int j = 0; j <= i; j++)
    {
        int f1 = 1;
        for (int k = 2; k <= i; ++k)
            f1 *= k;
        int f2 = 1;
        for (int k = 2; k <= i - j; ++k)
            f2 *= k;
        int f3 = 1;
        for (int k = 2; k <= j; ++k)
            f3 *= k;
        std::cout << f1 / (f2 * f3) << " ";
    }
    std::cout << "\n";
}
```