

TD n°2 – Langage C - 1^{ère} année

Exercice 1 : Considérons un programme contenant les déclarations et initialisations suivantes :

```
int i=8 ;
int j=5 ;
int k=9;
float x=0.005 ;
float y=-0.01 ;
char toto='c' ;
```

Déterminer la valeur de chacune des expressions suivantes :

- a) $i < j$
- b) $(i+j) \geq k$
- c) $(i+j) > (i+5)$
- d) $(i \geq 6) \parallel (j == 24)$
- e) $(i != 9) \&\& (j == 5)$
- f) $toto == 99$
- g) $2 * x + (y == 0)$
- h) $!(toto == 0)$
- i) $(x > y) \&\& (i > 0) \parallel (j < 5)$

Exercice 2 : Soit a un short non signé, de valeur 0xa2C3 en hexadécimal (base 16).

Ecrire le nombre binaire correspondant. Evaluer ensuite chacune des opérations binaires suivantes, en donnant à chaque fois le nombre binaire du résultat et sa valeur hexadécimale.

- a) $\sim a$
- b) $a \& 0x3F06$
- c) $a \wedge 0x3F06$
- d) $a \mid 0x3F06$
- e) $a \& \sim 0x3F06$
- f) $a \wedge \sim 0x3F06$
- g) $a \mid \sim 0x3F06$
- h) $a \gg 3$
- i) $a \ll 5$
- j) $a \& \sim a$
- k) $a \wedge \sim a$
- l) $a \mid \sim a$
- m) $a \& \sim 0x3F06 \ll 8$
- n) $(a \& \sim 0x3F06) \ll 8$
- o) $a \& \sim(0x3F06 \ll 8)$
- p) $a \wedge \sim 0x3F06 \ll 8$
- q) $(a \wedge \sim 0x3F06) \ll 8$
- r) $a \wedge \sim(0x3F06 \ll 8)$
- s) $a \mid \sim 0x3F06 \ll 8$
- t) $(a \mid \sim 0x3F06) \ll 8$
- u) $a \mid \sim(0x3F06 \ll 8)$

Exercice 3 : Soient i et j deux variables de type **int** valant respectivement 5 et 7, f et g deux variables de type **float** valant 5,5 et -3.25. Evaluer chacune des expressions suivantes :

- a) `i+=5`
- b) `f-=g`
- c) `j*=(i-3)`
- d) `f/=3`
- e) `i%=(j-2)`

Annexe :

Format de type char => 1 octet

Décimal	hexadécimal	binaire
0	0x00	0000 0000
1	0x01	0000 0001
2	0x02	0000 0010
3	0x03	0000 0011
4	0x04	0000 0100
5	0x05	0000 0101
6	0x06	0000 0110
7	0x07	0000 0111
8	0x08	0000 1000
9	0x09	0000 1001
10	0x0A	0000 1010
11	0x0B	0000 1011
12	0x0C	0000 1100
13	0x0D	0000 1101
14	0x0E	0000 1110
15	0x0F	0000 1111
16	0x10	0001 0000
17	0x11	0001 0001
18	0x12	0001 0010
19	0x13	0001 0011
20	0x14	0001 0100
21	0x15	0001 0101
22	0x16	0001 0110
23	0x17	0001 0111
24	0x18	0001 1000
25	0x19	0001 1001
26	0x1A	0001 1010
27	0x1B	0001 1011
....	
255	0xFF	1111 1111