

Ce sujet comprend 25 questions pour un total de 40 points.

## A Questions de cours

**Question 1** (2 points)

Qu'est ce qu'un masque ?

**Question 2** (3 points)

Quels opérateurs logiques utilise-t-on pour faire un masque ?

**Question 3** (2 points)

Quel est l'élément absorbant d'un ET ? d'un OU ?

**Question 4** (2 points)

Quel est l'élément neutre d'un ET ? d'un OU ?

**Question 5** (1 point)

Quel opérateur logique utilise-t-on pour forcer l'état d'un bit à 0 ?

**Question 6** (1 point)

Quel opérateur logique utilise-t-on pour forcer l'état d'un bit à 1 ?

**Question 7** (2 points)

Pour un valeur initiale de 8, quelle sera la nouvelle valeur si on décalage d'un bit à gauche ? à droite ?

**Question 8** (2 points)

Quelle est la différence entre un opérateur logique et bit à bit ?

## B Exercices

**Question 9** (1 point)

Si a et b sont définis par `int a = 0x5C;` et `int b = 0xD1;`, alors `(a & b)` vaut :

0x00    0x01    0x50    0x5C    0xDD    0xFF

**Question 10** (1 point)

Si a et b sont définis par `int a = 0x5C;` et `int b = 0xD1;`, alors `(a && b)` vaut :

0x00    0x01    0x50    0x5C    0xDD    0xFF

**Question 11** (1 point)

Si a et b sont définis par `int a = 0xC5;` et `int b = 0x6E;`, alors `(a | b)` vaut :

0x00    0x01    0x44    0x133    0xEF    0xFF

**Question 12** (1 point)

Si a et b sont définis par `int a = 0xC5;` et `int b = 0x6E;`, alors `(a || b)` vaut :

0x00    0x01    0x44    0x133    0xEF    0xFF

**Question 13** (2 points)

Quelle est la valeur de n après l'exécution des instructions suivantes ?

```
int n = 1; n = (n << 3) | 1;
```

0    3    7    1    8    9

**Question 14** (2 points)

Donner la valeur du masque dans l'opération `(a & masque)` pour lire seulement l'état du bit 3 :

`(a & 0x00)`    `(a & 0x03)`    `(a & 0x08)`    `(a & 0xF7)`

**Question 15** (2 points)

Donner la valeur du masque dans l'opération `(a & masque)` pour écrire le bit 3 à 0 :

`(a & 0x00)`    `(a & 0x03)`    `(a & 0x08)`    `(a & 0xF7)`

**Question 16** (2 points)

Donner l'opération pour écrire le bit 3 à 1 :

- (a & 0x04)    (a | 0x04)    (a & 0x03)    (a | 0x03)    (a | 0xF7)  
 (a & 0xF7)    (a & 0x01)    (a | 0x01)    (a & 0x00)    (a | 0x00)

**Question 17** (2 points)

Si a vaut 0x55, donner la valeur obtenue dans l'opération  $(a \& \sim(1 \ll 4)) \gg 4$  :

- 0x00    0x01    0x04    0xAA    0xFF

**Question 18** (1 point)

Dans l'opération  $(a \& \sim(1 \ll 4)) \gg 4$ , combien de bit(s) sont lu(s) ?

- 0    1    4    Tous    Aucun

**Question 19** (1 point)

Dans l'opération  $(a \& \sim(1 \ll 4)) \gg 4$ , quel bit est lu ?

- 0    1    4    8    Tous    Aucun

**Question 20** (1 point)

Dans une trame réseau, l'adresse IPv4 source est codée sur 32 bits par les valeurs hexadécimales : C0 A8 11 16. Quelle est alors la valeur de cette adresse IP en notation décimale pointée ?

- 192.168.11.16    192.168.11.10    192.168.17.10    192.168.17.22

**Question 21** (1 point)

En supposant que le masque est 255.255.255.0, Quelle sera alors l'adresse réseau en notation décimale pointée ?

- 192.168.17.0    192.168.17.22    0.0.0.22    192.168.0.0

**Question 22** (1 point)

En supposant que le masque est 255.255.0.0, Quelle sera alors l'adresse réseau en notation décimale pointée ?

- 192.168.17.0    192.168.0.22    0.0.17.22    192.168.0.0

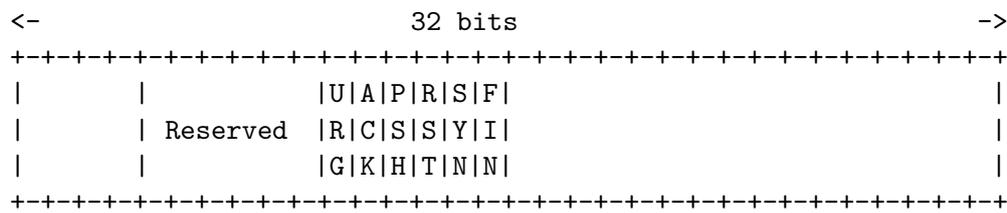
**Question 23** (2 points)

En supposant que le masque est 255.255.240.0, Quelle sera alors l'adresse réseau en notation décimale pointée ?

- 192.168.17.0    192.168.16.0    192.168.8.0    192.168.0.0

**Question 24** (2 points)

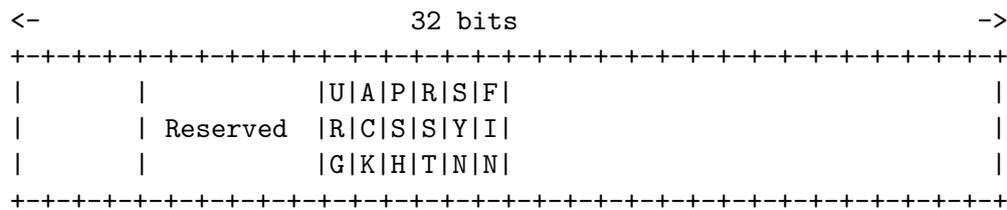
Dans une trame réseau, le champ *Flags* de l'en-tête TCP vaut en hexadécimal 0x18. Donner l'état des 6 *flags* ?



- URG=0 ACK=0 PSH=0 RST=0 SYN=0 FIN=0
- URG=0 ACK=1 PSH=1 RST=0 SYN=0 FIN=0
- URG=0 ACK=0 PSH=0 RST=1 SYN=1 FIN=0
- URG=0 ACK=0 PSH=0 RST=0 SYN=1 FIN=0
- URG=0 ACK=0 PSH=0 RST=0 SYN=0 FIN=1
- URG=1 ACK=1 PSH=1 RST=1 SYN=1 FIN=1

**Question 25** (2 points)

Donner la valeur en hexadécimal du champ *Flags* de l'en-tête TCP lorsque SYN=1 et ACK=1 ?



- 0x18
- 0x81
- 0x11
- 0x12
- 0x21