

1) HTML – Javascript (6 pts)

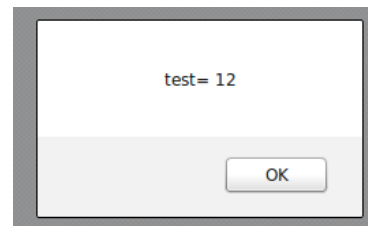
Soit le code suivant dans une page HTML:

```
<form name=formulaire action= "http://btsirisinfo.free.fr/jeu/compare.php" method =post>
<h1>Tapez votre nombre n( 1....100):</h1>
<input type=text name=monNombre ><br>
<input type =button value=Ok onclick=valide(monNombre.value) >
</form>
```

- 1.1) Donner
- l'adresse du serveur qui va traiter l'information;
 - le nom du fichier qui va recevoir l'information;
 - le nom de la variable transmise;
 - la méthode Http utilisée;
 - la signification de `onclick=Valide(MonNombre.value)`

1.2) Ecrire le code de la fonction **Valide (test)** qui affiche dans une fenêtre d'alerte le message "test=" suivi de la valeur de **test**

```
function Valide(test)
{///??
}
}
```



2) Réseaux (4pts)

2.1) Choisir 2 phrases correctes

- IP est encapsulé dans TCP
- TCP est encapsulé dans IP
- HTTP est encapsulé dans TCP
- TCP est encapsulé dans HTTP

2.2) Choisir 2 phrases correctes

- ICMP est encapsulé dans IP
- IP est encapsulé dans ICMP
- Ethernet est encapsulé dans ARP
- ARP est encapsulé dans Ethernet

2.3) Donner le PDU de la couche Transport

- paquet
- segment
- trame
- bits

2.4) Donner les 2 couches du modèle OSI correspondant à la couche *accès réseau* du modèle TCP/IP

- Application
- Transport
- Liaison de données
- Session
- Physique
- Présentation
- Réseau

4) Trame Ethernet (10pts)

A partir de la trame Ethernet ci-jointe, donner:

- 1) les adresses mac: destination et source en hexadécimal
- 2) les adresses IP destination et source en hexa puis en 4 nbres décimaux
- 3) les ports source et destination en hexa puis en décimal
- 4) la durée de vie TTL
- 5) l'état des bits SYN, ACK et FIN

Annexe

En-tête Ethernet :

6 octets	Adresse Physique (MAC) Destination
6 octets	Adresse Physique Emetteur (source)
2 octets	Type protocole réseau

En-tête Ip (type de protocole 0X0800) :

1er mot de 32 bits :

4 bits	Version
4 bits	Longueur en-tête en mots de 32 bits
8 bits	Type de service
-3 bits	priorité (0 normal, 1 lent)
-1 bit	délai (0 normal, 1 lent)
-1 bit	Vitesse de transmission (0 normal, 1 rapide)
-1 bit	Sécurité (0 normal, 1 haute)
-1 bit	
-2 bits	
16 bits	Longueur totale du paquet

2ème mot de 32 bits :

16 bits	Identification du datagramme
3 bits	Flags (gauche à droite)
	bit 0 toujours à 0
	bit 1 1: fragmenté, 0 : non fragmenté
	bit 2 Dernier fragment (1 : Oui, 0 : Non)
13 bits	Offset du fragment

3ème mot de 32 bits :

8 bits	Temps de vie: TimeToLive
8 bits	Type de protocole
16 bits	Checksum en-tête

4ème mot de 32 bits :

32 bits	Adresse IP Emetteur
---------	---------------------

5ème mot de 32 bits :

32 bits	Adresse IP Destinataire
---------	-------------------------

6ème mot de 32 bits (éventuel voir longueur de l'entête dans 1er mot):

x bits	Options
y bits	Bourrage

En-tête TCP :Placé après un en-tête IP (type de protocole :0x06)

16 bits	Port source
16 bits	Port destination
32 bits	Offset depuis début de la transmission
32 bits	Accusé de réception
4 bits	La taille de l'en-tête TCP en nombre de mots de 32 bits.
6 bits	Réservé (bits à 0)
6 bits	Bits de contrôle voir note ci-dessous
	Infos diverses, checksum, bourrage

Note:Bits de contrôle (de gauche à droite):

URG: Pointeur de données urgentes

significatif

ACK: Accusé de réception significatif

PSH: Fonction Push

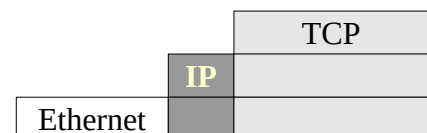
RST: Réinitialisation de la connexion

SYN: Synchronisation des numéros de

séquence

FIN: Fin de transmission

Encapsulation des couches:



n°octet

TRAME ETHERNET

0000	00 07 ac dc ac dc e1 1e ad bb ca 55 08 00 45 00
0010	00 30 18 17 40 00 80 06 0e 5f ac 20 10 05 0a 0b
0020	30 15 04 01 00 50 63 9d d4 35 00 00 00 00 70 02
0030	40 00 31 f9 00 00 02 04 05 b4 01 01 04 02