

Module : AJEL, Niveau :2LMD, Chargée de module : Mme Hattab Dalila.

Question (1) : (10 points)

A :1,3 (1 point)	B :1,2,3,4 (1,5 point)	C :2,3,4 (1 point)	D :1,5 (1 point)
E :2,3 (1 point)	F :1,2,3 (1 point)	G :1,4,3 (1 point)	H :1,3 (1 point)
I : 1,2,4 (1,5 point)			

Question (2) : (1.5 point+ 1.5 point+1 point)

	Droit d'auteur	Brevets	Marques
Objet de la protection	Les œuvres littéraires et artistiques  (auxquelles sont assimilés les programmes d'ordinateur).	Les inventions, c'est-à-dire les solutions techniques à des problèmes techniques  (inventions mises en œuvre par programme informatique).	Les signes (mots, logos, etc.) permettant de distinguer un produit ou un service d'un autre produit ou service.
Etendue de la protection	Les éléments originaux de la forme.  (le code)	Les procédés techniques  (les fonctionnalités)	L'usage du signe pour distinguer un ou plusieurs produits ou services déterminés
Conditions	- mise en forme - originalité	- nouveauté - activité inventive - application industrielle	- signe pouvant être reproduit graphiquement - signe distinctif - disponibilité du signe (pas identique ou similaire à un signe utilisé pour produit ou service identique ou similaire) - restrictions diverses
Durée	70 après la mort de l'auteur	20 ans maximum	10 ans - renouvelable indéfiniment

Question (3) : Définition de licence : (1 point)

La licence est le contrat qui donne à un utilisateur, sous condition, le droit d'utiliser un produit logiciel. Il pourrait exister autant de licences que de logiciels. Nous allons procéder ici à un classement par grandes familles de licences, en fonction des contraintes qu'elles imposent. Chaque type de licence apporte un certain nombre d'avantages et d'inconvénients à l'éditeur et à l'utilisateur. Les contrats couvrent généralement quatre aspects :

- Le droit d'utilisation ;
- Le droit de redistribution ;

- Le droit de modification ;
- La contrepartie financière.

## 2.2 les différentes classes des logiciels selon la licence en :

1. Les logiciels propriétaires : un logiciel dont la licence ne permet que l'utilisation. Celle-ci est généralement règlementée et limitée : il est souvent nécessaire d'acheter une licence par utilisateur, ou d'acheter des licences de sites. **(1 point)**
2. Les logiciels dits « du domaine public » : Le domaine public désigne l'ensemble des créations intellectuelles pour lesquelles les droits d'exploitation liés au droit d'auteur sont expirés. **(1 point)**
3. Les gratuiciels (*freewares*) : *Les gratuiciels sont un sous-ensemble de logiciels propriétaires. Ils partagent avec eux les mêmes restrictions pour l'utilisateur. Ils ne sont diffusés que sous forme d'exécutables binaires, et non pas sous forme de code source. Il est interdit de les modifier.* **(1 point)**
4. Les partagiciels (*sharewares*) : *Les partagiciels sont des logiciels propriétaires, disponibles en versions d'évaluation. Si l'utilisateur est satisfait du produit, et qu'il souhaite continuer à l'utiliser, il doit s'acquitter du prix de la licence.* **(1 point)**
5. Les logiciels libres : Les logiciels libres sont un ensemble de logiciels fondés sur un tout autre modèle économique que l'ensemble des logiciels propriétaires que nous avons vus jusqu'à présent. Un logiciel libre est un logiciel dont il est possible de modifier les sources à volonté, et de redistribuer les modifications sans contrainte. **(1 point)**

Corrigé :

### Vertex shader

```
varying vec3 FragPos;
varying vec3 normale;

void main()
{
    /* FragPos = Position du vertex dans l'espace Monde */
    FragPos = gl_Vertex;

    /* calculer le vecteur normal dans les coordonnées globales */
    vec3 normale = normalize(gl_NormalMatrix * gl_Normal);
    /* projection sur l'écran */
    gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
}
```

3.0

### Fragment shader :

```
uniform vec3 lightPos; /* position de la lumière (S) */
uniform vec3 viewPos; /* position de l'observateur (v) */
uniform float f; /* shininess */
uniform vec3 La, Ka, Ld, Kd, Ks, Ls;

varying vec3 FragPos;
varying vec3 normale;

void main()
{
    /* Ambient */
    Vec3 ambient = Ka * La;

    /* Diffuse */
    vec3 S = normalize(lightPos - FragPos);

    vec3 N = normalize(normale);

    float cosT = max(dot(S,N), 0.0);

    vec3 diffuse = Kd * Ld * cosT;

    /* Specular */
    vec3 V = normalize(viewPos - FragPos);

    /* calcule du vecteur H = (s+v / ||s+v|| ) */
    vec3 H = normalize(S + V);
    float spec = Ks * Ls * pow(max(dot(N, H), 0.0), f);

    gl_FragColor = vec4(ambient + diffuse + spec, 1);
}
```

4.0