

# Plagiat et Honnêteté intellectuelle

## CPE

- Tout plagiat (copie sans citation de sources) est interdite sur l'ensemble des documents rendus.
- L'honnêteté intellectuelle consiste à citer les sources directes ou indirectes d'inspiration qui ont permis de réaliser un travail.

*Règle fondamentale* : **Citez vos sources.**

Il est essentiel que, lors du déroulement des examens, comme lors de toutes les activités pédagogiques, les élèves fassent preuve du comportement responsable et professionnel attendu de futurs ingénieurs, dans un souci permanent d'éthique et d'honnêteté.

Citation du règlement des études de CPE :

(<https://e-campus.cpe.fr/mod/resource/view.php?id=36>)

# 1 Dénomination

Le plagiat consiste à utiliser tout ou partie du travail de quelqu'un d'autre sans le mentionner.

*Remarque sur la quantité de la copie* : La quantité de plagiat peut varier de l'ensemble d'un document à une seule ligne (code et/ou texte). Il n'y a pas de notion de *quantité admissible*.

*Remarque sur les licences* :

- Utiliser le contenu d'un document *libre* sous licence *libre* telle que GPL <sup>1</sup> dans un travail implique que votre travail passe en totalité sous licence *libre* également (ex. Utilisation de certaines images Wikipédia).  
C'est à dire : distribution intégrale du code source avec l'exécutable, et mention de la licence libre sur chaque fichier.
- Utiliser le contenu d'un document sous copyrights sans permission explicite de l'auteur est interdit et peut entraîner des sanctions pénales et financières.

Dans le cadre de travaux d'étudiants, il est toléré de pouvoir utiliser des documents sous licence libres en les citant, sans avoir à placer une notion de licence sur ce travail. L'utilisation de documents sous copyright cités explicitement est généralement tolérée, mais vous engage à titre personnelle de vérifier que la copie n'est pas explicitement interdite.

**Dans tout les cas, l'absence de citation de la source est interdite et pénalisée.**

## 2 Utilisation de ressources existantes

- Il n'est pas interdit d'utiliser Wikipedia, ni de le citer.
- Il n'est pas interdit de recopier mot pour mot un texte existant à condition de le citer explicitement.
- Il n'est pas interdit d'utiliser le même code, le même paragraphe, ou les mêmes résultats que votre voisin à condition de citer explicitement la source.

## 3 Notation dans le cadre scolaire

- La copie sans citation est interdite et sanctionnée.
- La copie avec citation est autorisée mais ne rapporte pas de points.
- La copie avec citation ainsi qu'une analyse ou une extension supplémentaire commence à rapporter des points.

Vous pouvez donc utiliser à bon escient le travail d'autres personnes citées en le complétant et en l'analysant.

---

1. <http://www.gnu.org/licenses/gpl.html>

## 4 Exemples de citations

*Note préliminaire* : Il vaut mieux citer trop de références que pas assez.

### 4.1 Version professionnelle :

Nous implémentons l'approche décrite par Cockburn [C05].

(à la fin du document) **Bibliographie** :

[C05] Alistair Cockburn. *Agile Software Development*. Addison-Wesley Professional, 2005.

### 4.2 Citation Wikipedia :

Comme le décrit la page Wikipedia, les tests d'intégration *ont pour but de valider le fait que toutes les parties développées indépendamment fonctionnent bien ensemble de façon cohérente..*

Nous utilisons le code du tri par insertion donné sur la page Wikipedia ([http://fr.wikipedia.org/wiki/Tri\\_par\\_insertion](http://fr.wikipedia.org/wiki/Tri_par_insertion)):

```
procédure TriInsertion(tableau T, entier n)
  pour i de 2 à n
    x = T[i]
    j = i
    tant que j > 1 et T[j - 1] > x
      T[j] = T[j - 1]
      j = j - 1
    fin tant que
    T[j] = x
  fin pour
fin procédure
```

### 4.3 Citation Wiki (2) :

Il nous a paru intéressant de coder une courbe de Bézier en langage Python. Pour cela, nous nous sommes inspirés du code présent sur ce Wiki :

[http://rosettacode.org/wiki/Cubic\\_bezier\\_curves#Python](http://rosettacode.org/wiki/Cubic_bezier_curves#Python)

Que nous avons ensuite intégré dans notre visualiseur. Notez que nous avons généralisé ce code à des courbe de degrés 4 également (voir notre annexe).

### 4.4 Citation d'un site sous copyright :

Pour implémenter notre gestion mémoire par allocation dynamique, nous nous inspirons du code fourni sur le *site du zero* [SZERO]

(à la fin du document) **Bibliographie** :

[SZERO] [http://www.siteduzero.com/tutoriel-3-14061-1-allocation-dynamique.html#ss\\_part\\_3](http://www.siteduzero.com/tutoriel-3-14061-1-allocation-dynamique.html#ss_part_3), auteur : Mathieu Nebra.

Pour réaliser notre implémentation de liste chaînée, nous avons utilisé le code trouvé sur le site *developpez.com*.

<http://nicolasj.developpez.com/articles/listedouble/>

## 4.5 Inclusion d'une image sous copyright :

Nous avons trouvé sur internet une image illustrant très bien la méthodologie d'extrême programming. L'image X est issue du site <http://xprogramming.com>

## 4.6 Citation d'un collègue :

Suite aux discussions avec notre binôme voisin (Jean Moulin et Bernard Dupont), nous avons décidé de construire une fonction d'initialisation de notre tableau séparée du reste de l'algorithme.

## 4.7 Citation d'un collègue (2) :

Nous n'avons pas réussi à coder cette partie du TP/projet. Afin d'avancer et de pouvoir réaliser la partie suivante, nous avons tout de même demandé au binôme (Jean Moulin et Bernard Dupont) de nous fournir leur code.

Nous avons tout de même noté que cette fonction stockait en dur le nombre de cas à vérifier. Nous avons tenté d'améliorer ce code en introduisant un nombre de cas variable. Nous avons testé l'application de la version améliorée dans le cas où l'on considère une grille de jeu plus importante.

## 4.8 Citation d'un collègue dans du code :

Voir figure. 1.

```
//Code de creation de tableau_2d
// Rem. Code repris du binome Jean Dumont et Francois Rousseau
//   car permet de factoriser nos appels a malloc.
// Note: Nous aurions pu utiliser l'instruction calloc pour
//   initialiser en meme temps qu'allouer.
float* cree_tableau_2d(int taille_x,int taille_y)
{
    float *tableau=NULL;
    tableau=malloc(taille_x*taille_y*sizeof(float));

    if(tableau==NULL)
    {printf("Erreur allocation tableau %dx%d\n",
        taille_x,taille_y);exit(1);}

    //initialisation valeurs a 0
    int k=0;
    for(k=0;k<taille_x*taille_y;++k)
        tableau[k]=0.0;
    return tableau;
}

//Code de liberation memoire de tableau_2d
// Rem. Code repris du binome Christelle Beauvois
//   et Francois Aura (groupe C).
// Leur code est tres robuste (detection des cas d'erreurs)
// et nous a permis de continuer a avancer dans notre
// projet sans segmentation fault.
void supprime_tableau_2d(float **tableau_a_supprimer)
{
    //verification des entrees
    assert(tableau_a_supprimer!=NULL);
    assert(*tableau_a_supprimer!=NULL);

    free(*tableau_a_supprimer);
    *tableau_a_supprimer=NULL;

    //contrat de sortie
    assert(*tableau_a_supprimer!=NULL);
}
```

FIGURE 1 – Exemple de citation de collègues dans du code.

## 5 Sanctions

Toute détection d'un plagiat quelconque est signalée automatiquement à la direction des études de CPE.

*Notez que la sanction de plagiat aura lieu sur toute forme de copie et sur n'importe quelle quantité. Que ce soit pour une copie intégrale d'un TP/projet, ou pour la copie d'une seule ligne d'un site internet, ou d'une seule ligne du TP voisin. Même une seule ligne copiée nécessite la citation adéquate.*

Le règlement des études

(<https://e-campus.cpe.fr/mod/resource/view.php?id=36>)

indique que :

Le plagiat en examen ou lors de rendu de travaux évalués peut faire l'objet d'un conseil de discipline dont les sanctions peuvent être :

- un avertissement,
- un blâme,
- une exclusion temporaire,
- une exclusion définitive.

De plus, quand il s'agit de fraude, plagiat, ou tentative de fraude, toute sanction invalide le module concerné.