

# EFREI L1 TOPC

## PJ 1 - Application des bases de l'algorithmique

*Conjecture de Teller*

*v3 du 29 octobre 2016 - Franck Lepoivre & Patrick Teller*

### Équipes, collaboration

**Enregistrement des équipes** : <https://goo.gl/QODLB0>

Travaux d'analyse, conception et programmation, et soutenances sont effectués par équipes de deux étudiants au plus (équipes TP + PJ).

Ces équipes sont fixées d'après vos enregistrements et pour les touristes, par attribution d'office. Vous avez jusqu'à **vendredi 18 novembre** pour vous enregistrer

En cas de souci au sein de votre équipe, sollicitez votre enseignant ou votre responsable de cours pour tenter de résoudre le problème à travers un entretien à trois.

**Document détaillé à ce sujet** : <https://goo.gl/nktPUO>

### Planning du projet / délais de réalisation et de rendu

Les dates clé :

- Distribution du sujet de projet le **23 octobre 2016**.
- TP de lancement de projet les **25 octobre ou 3 novembre 2016** suivant les groupes.
- Soutenances de projet les **29 novembre ou 1<sup>er</sup> décembre 2016** suivant les groupes.
- Limite pour la livraison des travaux les **29 novembre ou 1<sup>er</sup> décembre 2016** suivant les groupes.

### Conseils

#### **Méthode : prendre du recul**

Foncer tête baissée dans la réalisation est toujours une mauvaise stratégie. Une telle démarche est opposée à ce que l'on attend d'un ingénieur. Pensez-y, vous serez bientôt un jeune professionnel.

Il est important d'analyser les problèmes posés et de concevoir vos algorithmes avant de les faire valider (par vos camarades, par vos enseignants, etc).

Il est important de vous donner des objectifs fonctionnels, de les prioriser (must have > should have > nice to have), de vous organiser dans le temps en vous répartissant le travail. Ne négligez pas les tests unitaires et tests d'intégration.

## **Culture du résultat**

On oppose classiquement obligation de moyens et obligation de résultat.

Par le premier, on entend l'obligation de fournir ce qu'il faut d'effort, sous-entendu avoir donné de la sueur, de la souffrance, du temps pour produire quelque chose.

Par le second, on entend l'obligation de fournir quelque chose qui marche sans s'intéresser aucunement à la manière dont cette chose a été produite.

La culture du résultat signifie que votre rétribution (la note) est fondamentalement lié à ce que vous produirez, et aucunement au nombre d'heures que vous y aurez passé, pas plus qu'au nombre de difficultés que vous aurez surmontées.

Si le malin produit la même chose que vous en y passant 10 fois moins de temps, au lieu de le jalouser et de faire valoir votre effort, cherchez plutôt à voir en quoi il a été plus performant que vous et en tirer parti pour améliorer votre démarche.

Ce qui vient d'être dit n'est en aucun cas une invitation à tricher. Nous serons absolument intransigeants avec la triche qui est une version dégradée de l'intelligence.

## **Exemple de traduction pratique**

Avec un programme qui ne compile pas ou plante vous ne pouvez espérer dépasser 10/20.

Par conséquent, vous devez gérer le risque d'investir beaucoup de moyens pour un petit résultat si votre programme ne compile pas ou plante :

1. mieux vaut moins de fonctionnalités qui marchent que trop de fonctionnalités qui ne marchent pas.
2. faites beaucoup de tests et provisionnez suffisamment de temps pour la mise au point et la finition de votre application (corrections de bugs, tests fonctionnels, amélioration de l'expérience utilisateur).

## **La soutenance est un exercice de communication**

Il faut que votre communication orale soit claire, concise, efficace.

Pensez à contextualiser (planter le décor) en introduisant le sujet dont il est question et en montrant que vous le maîtrisez. Annoncez que vous allez faire une démonstration et conduisez la efficacement en annonçant, à chaque étape de celle-ci, ce que vous allez montrer avant de le montrer. Soyez honnête sur ce que vous avez pu réaliser ou pas par rapport au cahier des charges (le sujet).

## **Livrable**

### **Contenu du livrable**

Votre livrable se présente sous la forme d'une archive contenant :

- vos fichiers sources sans les fichiers binaires (pas de .o ni .exe) ;

- les images et autres ressources intégrées dans vos pages ;
- votre rapport, impérativement au format PDF ;
- un fichier README.txt donnant un compte rendu honnête et fiable, en 10 à 20 lignes, de votre niveau d'avancement par rapport au travail demandé et des principales difficultés rencontrées + les consignes d'installation de librairies tierces si vous en utilisez.

## Rapport

- Généralités
  - Un sommaire détaillé avec pages numérotées, un minimum d'effort de design et de mise en page.
  - Votre analyse du problème posé par le cahier des charges.
  - Un état de l'art synthétique du sujet, et notamment des solutions standards qui répondent au problème posé.
  - La présentation des grandes lignes de votre solution au problème posé : architecture générale de votre application.
  - Présentation des structures de données et algorithmes clés :
    - composition de la structure de donnée, codage de l'information avec la justification des choix effectués.
    - principe de l'algorithme (expliquer l'idée en termes simples et intelligibles) puis sa spécification en langage algorithmique (la présence de C dans votre rapport serait hautement préjudiciable).
  - Une conclusion qui restitue votre expérience projet, ce que vous avez appris, les difficultés rencontrées et comment vous les avez surmontée, ce que vous essaieriez d'améliorer au prochain projet.
- En particulier pour ce projet :
  - Prendre à peu près la même trame que celle du sujet pour diviser les principales sections de votre rapport.
  - Présenter à chaque fois, de manière synthétique, les résultats empiriques obtenus, expliciter les algorithmes employés en langage algorithmique, donner vos arguments et démonstrations mathématiques pour les questions marquées ET dans le sujet.
  - Votre rapport doit présenter une synthèse en une quinzaine de pages et peut être complété par des annexes (par exemple, algorithmes détaillés dont vous ne présentez que quelques extraits dans la synthèse qui précède, idem avec résultats empiriques et démonstrations).

## Livraison

Le chef de projet du binôme (ou le monôme) remettra son **livrable au format ZIP ou RAR** au plus tard au plus tard à 23:59 le soir de la soutenance à l'adresse de l'enseignant en charge de son évaluation :

Gp	Enseignant	Adresse de remise
A	Kamel Chabchoub	chabchoub.efrei@gmail.com
	Ithéri Yahiaoui	itheri.yahiaoui@univ-reims.fr
B	Christopher Marshall-Breton	christopher.marshallbreton@gmail.com
	Kaïs Klai	kais.klai@lipn.univ-paris13.fr
C	Jules Lepoivre	jlepoivre@saki-studio.com
	Kaïs Klai	kais.klai@lipn.univ-paris13.fr
D	Maxime Marion	livraison-11@boiteaspam.com
	Halim Djerroud	hdd@ai.univ-paris8.fr
E	Karim Lahlou	karim.lahlou@efrei.fr
	Lamia Benamara	lamia.benamara@platypus.academy
F	Christopher Marshall-Breton	christopher.marshallbreton@gmail.com
	Kamel Chabchoub	chabchoub.efrei@gmail.com
PL	Kathia Chenane	chenane.kathia@gmail.com
	Mouna El Attar	mimimea2@yahoo.fr

Une seule remise par équipe PJ. Par prudence, si vous êtes deux, mettre votre binôme en copie de votre email de remise.

### **Votre enseignant vous enverra un accusé de réception opposable.**

Si vous n'avez pas reçu celui-ci sous trois jours, il vous appartient de vous en inquiéter.

L'objet de votre email comme le nom de votre fichier d'archive sont :

**EFREI L1 TOPC | PJ 1 | Livrable** suivi du nom de votre équipe tel que vous l'avez enregistré au moment de sa constitution.

Par exemple : **EFREI L1 TOPC | PJ 1 | Livrable - Les pieds nickelés**

Ces instructions sont complétées par les consignes suivantes.

**Ne pas les respecter vaut des points de pénalité.**

Lien direct : <https://goo.gl/fxz2tC>.

# Évaluation

## Principaux critères d'évaluation

Les principaux critères d'évaluation sont :

- Communication :
  - La qualité de l'oral, dont en particulier de la démonstration.
  - La qualité du rapport, tant sur le fond que sur la forme.
- Exécutable :
  - La couverture fonctionnelle, cf. ce qui était demandé.
  - La lisibilité des interfaces.
  - La facilité d'utilisation, l'ergonomie des interactions.
- Code :
  - La qualité des structures de données et des algorithmes.
  - La qualité d'écriture du code (indentations, commentaires, choix des identifiants, etc).
- Rigueur :
  - Respect des consignes de livraison, en particulier de la deadline.
  - Qualité du README

La moyenne pour chaque groupe tourne autour de 12.

Il n'y a pas de contrainte d'écart type et les meilleures équipes peuvent avoir 20.

## Barème

Barème						
2	10	PRG modulé par %couv fonctio nnelle	6	EXE	Compilation et exécution	
2					Lisibilité des traces d'exécution	
2					Facilité d'interaction	
2				4	CODE	Qualité des structures de données et des algorithmes
2						Qualité d'écriture du code (indentations, commentaires bien dosés, choix des identifiants, etc)
3	3		3	RIGUEUR	Respect des consignes de packaging (objet de l'email, nom de l'archive, etc)	
0,5	7	Comm. / démo.	4	RAPPORT	Forme : structure	
0,5					Forme : mise en page, présentation, design	
1					Forme : orthographe, expression écrite	
2					Fond : recul, pertinence, synthèse	
3					3	SOUTENANCE
20	20		20	TOTAL		
Note : (exe + code) x couv + rapport + soutenance + rigueur						

La note moyenne, dans chaque groupe de TD est comprise dans la fourchette 10 à 15. Il n'y a pas de limite d'écart type et les meilleurs travaux peuvent recevoir 20 comme les pires recevoir 0.

## Des questions ?

Un formulaire pour les poser qui aidera à composer une FAQ projet :

<https://goo.gl/HCj1uz>