

# Validation d'algorithmes

Chapitre X

Présentation :  
Déroulement du cours

- × Chapitre 1 : Variants, invariants
- × Chapitre 2 : Exemples
- × Chapitre 3 : Tests boîte blanche
- × Chapitre 4 : Complexité cyclomatique
- × Chapitre 5 : Machines de Mealy
- × Chapitre 6 : Tests boîte noire
- × Chapitre 7 : Sémantique opérationnelle
- × Chapitre 8 : Logique de Floyd-Hoare



- ⊗ Savoir se poser la question « Pourquoi mon programme fonctionne ? »
- ⊗ Introduction aux techniques pour vérifier un algorithme
- ⊗ Formaliser le fonctionnement d'un programme.
- ⊗ Comprendre certaines preuves haut-niveau
- ⊗ Découvrir différents types de tests
- ⊗ Savoir bien tester
- ⊗ Abstraire un programme

Ces règles ne devraient même pas être citées, mais vu que la promo précédente c'est comporté de manière puérile. . .

- × **Comportez-vous en adulte !**
- × Vous êtes là pour suivre le cours, si vous voulez discuter, ne venez pas en cours.
- × On arrive à l'heure, ou en cas de retard on reste discret.
- × On respecte les règles de courtoisie de base.
- × On prend des notes de cours.
- × On relit son cours avant le TD suivant.
- × Si vous êtes absents, prévenez votre responsable de TD et moi-même.

- ⊗ N'hésitez pas à poser des questions si un point du cours vous semble obscur.
- ⊗ Les TD sont une occasion de poser les questions sur les parties précédentes.



- ⊗ 50% en CC.
  - ⊗ Une note de rendus de TD (voire en TD pour les détails).
  - ⊗ Partiel (seulement si présentiel possible). [Date à valider, le 14 février ?]
- ⊗ 50% à l'examen final.
  - ⊗ Tout le programme sera à l'examen final, les séances de TD sont ce qui vous y prépare le mieux, ne négligez pas ce temps qui peut facilement réduire vos révisions.
  - ⊗ **Si possible** Devoir classique.
  - ⊗ **Sinon** Projet individuel. L'idée est de mettre en pratique toutes les méthodes de validation/test vues en cours sur un programme réel (exemple de programme réel : zip, openssl, libcairo, libregex).

