

MIC

- Réussir son S1 de 3MIC Starter Pack™
- Réussir son S2 de 3MIC-IR Starter Pack™
- Réussir son S2 de 3MIC-MA Starter Pack™

Réussir son S1 de 3MIC Starter Pack™

Les bases à avoir avant d'arriver en cours:

•

Les liens ressources

- La page **Cyber-Aide** : [lien](#)
- [Drive GAMMA](#)
- Le **Club Info** organise des **tutorats** notamment en Git (utile pour le S2 et le projet du S1)
- Le site [annales](#) contient encore quelques ressources intéressantes : [lien](#).
- Un google drive assez complet (merci Théo) : [lien](#).
- D'autres ressources étudiantes sont disponibles :
 - Le site de Louis : [lien](#).

Conseils plus pratiques

Général:

Tronc commun :

UF - Analyse Numérique, Optimisation, Chaînes de Markov :

Analyse Numérique et Optimisation :

Pendant les TPs, utilisation de Python en Jupyter Notebook. Il est donc conseillé de revoir ses cours pour ne pas perdre de temps (notamment les affichages la matplotlib). Aussi, les TPs sont indispensables pour bien connaître la seconde partie du cours, donc ne les prenez pas à la légère,

voir les refaire.

Chaînes de Markov :

UF - Système d'exploitation, Réseaux, Base de données :

Système d'exploitation :

Revoir l'architecture matérielle aidera beaucoup : une partie du cours est en commun.

*Disclaimer : Si ça n'a pas été corrigé, les contrôles ne se reposent pas sur le cours, mais sont des copier-coller des contrôles des années précédentes (donc des cours en partie plus vus). Donc **faites les annales**, sinon vous n'aurez pas une partie des points :)*

Réseaux :

Site du MIT avec des vidéos très bien expliquées et des schémas/questions de cours et d'application :

https://gaia.cs.umass.edu/kurose_ross/lectures.php

Base de données 2 :

Bien connaître sa BDD 1 n'est pas obligatoire, mais conseillé.

La note ne se fait que par projet, mais attention ! Contrairement à la BDD1, corriger un bug ou un manque peut prendre extrêmement longtemps. Il est conseillé de passer un maximum de temps sur la **partie théorique** de conception des bases avant d'implémenter.

Pour les fans d'IA, utiliser ChatGPT peut être pratique, notamment pour implémenter parfois des dizaines de lignes avec des valeurs aléatoires pour des tests. Néanmoins, n'en abusez pas, la BDD2 sera utile pour les prochains semestres, donc plus vous pratiquez, mieux vous maîtriserez. :)

Enfin, vous pouvez utiliser **git** (sur **GitEtud**) ou **Visual Studio Code** avec l'option **Live Share** pour pouvoir mettre vos tables en commun.

Option Mathématiques Appliquées :

UF - EDO et leur résolution numérique :

UF - Calcul intégral et probabilités :

Option Informatique et Réseaux :

UF - Base de données et programmation web :

Base de données 1 :

Programmation Web :

Vous y verrez l'HTML, le CSS et le Javascript. Pour ce dernier, revoir vos cours de C peut aider.

Nous vous recommandons d'utiliser WebStorm de la suite [JetBrains](#)

Pour la mise en commun : Utiliser **GitEtud** peut être très pratique pour le projet, donc n'hésitez pas à apprendre les bases (Créer un projet, commit, push, merge). Sinon, **Visual Studio Code** avec l'option **Live Share** peut fonctionner pour les moins vaillants.

Pour voir votre projet : Tous les étudiants de l'INSA ont accès à un **hébergement gratuit** sur le **serveur etud**. Si vous avez besoin d'aide, vous pouvez poser toutes vos questions sur le **serveur du Club Info**. Utile pour le rendu du rapport. (nécessitant un lien pour accéder à vos pages). Il y a aussi un preview sur git.

Sinon, pour pouvoir avoir le rendu en direct, vous pouvez **ouvrir le fichier main.html dans votre navigateur** ou utiliser l'extension **Live Server** sur **Visual Studio Code**.

UF - Outils théoriques pour l'informatique :

Algorithmique Avancée :

Vous étudierez la complexité des algorithmes, et certains théorèmes pour mieux comprendre la complexité des algorithmes. Mais ceci n'est pas un cours de programmation, vous ne toucherez donc pas aux ordinateurs pendant ce cours. Il n'y a pas de pré-requis particuliers.

Expressions Régulières :

L'examen de cette matière est facultatif, et vous pouvez valider cette matière si vous venez au TP (qui n'est pas noté). Cependant, essayez quand même de travailler cette matière, car elle vous sera forcément très utile par la suite dans vos scripts ou pour chercher quelque chose dans de longs fichiers de code. (un peu l'équivalent d'un contrôle F, mais en beaucoup plus poussé). Il n'a pas non plus de prérequis à avoir.

Programmation linéaire :

Il n'y a pas de prérequis à avoir. C'est une matière pas très difficile qui permet d'avoir une bonne note si on travaille bien. Pour cela, je recommande de suivre attentivement les TD, car cette

matière peut permettre d'avoir une bonne note pour rattraper Théorie de l'information et codages, qui est une matière très incertaine où on peut avoir de très mauvaises notes, même en ayant compris.

Théorie de l'information et codages :

Réussir son S2 de 3MIC-IR

Starter Pack™

Le second semestre de 3 MIC IR est le premier semestre où la notion de projet commence à prendre une place importante. En effet, plusieurs bureaux d'études sont prévus au cours du semestre et les projets comptent pour une part importante dans la notation.

Les bases à avoir avant d'arriver en cours :

- Avoir un niveau élémentaire en C (maîtriser les parties 1, 2 & 3 de [ce cours Open Classroom](#) ou se référer au cours du premier semestre)
- Bien maîtriser les réseaux TCP/IP (voir cours du premier semestre ou [ici cours en Français OpenClassroom](#).
Meilleure solution pédagogiquement, mais en anglais : [Ce site](#) comprend des vidéos super bien expliquées en anglais. Ce cours du MIT est souvent utilisé comme base pour les cours des profs)
- Une première expérience en *algorithme de recherche*, en *java* ou avec *gdb/assembleur* peuvent simplifier le semestre.
- Certains cours se basent sur de la [complexité](#)

Pour les étudiants ayant fait un premier semestre à l'étranger, pensez à vous mettre à jour en C et en réseaux si vous n'avez pas eu de cours ! Le retard dans ces domaines peut être très pénalisant. Inutile de rattraper les maths et les BDD cependant.

Les liens ressources

- La page **Cyber-Aide** : [lien](#)
- La page **Gamma** : [lien](#)
- Le site annales contient encore quelques ressources intéressantes : [lien](#).
- Un google drive assez complet (merci Théo) : [lien](#).
- D'autres ressources étudiantes sont disponibles :
 - Le site de Louis : [lien](#)

Conseils plus pratiques

Général

Le semestre n'est pas très lourd en travail dans la mesure où l'on ne fait pas tout à la fin. Respectez les délais indiqués par les profs (en particulier en BE Réseaux) et n'hésitez à prendre un weekend de travail si vous commencez à accumuler du retard. Pensez aussi à vous renseigner sur la forme du rendu : ce n'est pas toujours un rapport ! C'est toujours mieux de ne pas travailler pour rien...

INFORMATIQUE, OUTILS MATERIELS ET NUMERIQUES

Assembleur

Bien connaître son cours, les syntaxes particulières (sur les retours, sur les accès en mémoire etc). En 24-25 les TD étaient en premiers. C'était dur à appréhender mais le BE CHTI qui se fait en ARM aide beaucoup pour les TP qui sont bien après. Les TP de POM (et CM) sont super bien faits merci Bit-Monnot.

Sécurité

Il y a 2 TP de sécurité vers la fin de l'année il faut vraiment bien les comprendre puisque l'examen est jusqu'ici un examen de sécurité où il faut bien comprendre l'assembleur intel et surtout gdb. Nous n'apprenons pas vraiment comment bien utiliser gdb, faites le de votre côté (les jump surtout). Les annales sont normalement disponible et globalement tous les ans il y a un digicode (cette année c'était le flag 3) dans lequel il vaut mieux regarder la rom extraite, qui permet de comprendre comment marche le test du digicode (surtout quel chiffre le digicode vérifie en premier). Il y a tout le temps une fonction où l'on doit mettre un breakpoint après qu'elle ait chargée un mot de passe / login à leur adresses.

BE CHTI

Il faut quelques notions de signal mais une fois matlab passé c'est surtout pas mal d'assembleur, KEIL et de faire du marcher du code C avec de l'assembleur. Pour l'examen soyez bons en ARM (faire un for, un if et while c'est déjà pas trop mal, faire une division en décalant les bits..) et il ne devrait pas y avoir de problème (un petit de signal mais pas beaucoup et en rapport avec ce qu'on a fait en partie 1).

POO

Les TDs sont sur [le site de Le Botlan](#) . N'hésitez pas à jeter un coup d'oeil à votre code avant l'examen pour vous remémorer les concepts. Le plus dur, c'est de comprendre ce qu'on vous demande mais si vous avez compris les concepts de la POO ça ne présente pas de difficulté particulière. Cette année (24-25) nous n'avons pas eu accès à des ide comme vscode pour faire les

getter / setteur... On espère que ça changera mais en fait bien les annales globalement vous ne risquez rien il faut juste bien connaître la structure de l'annale pour être rapide (on manque de temps si on ne connaît pas la structure).

Graphes

Normalement, il y a droit à une fiche pour l'examen. C'est tous les ans à peu près la même chose, sachez bien appliquer les algorithmes vus en cours et ayez fait quelques annales pour voir le genre de questions un peu difficile qui tombent aux exams.

BE Graphes

Attention, la prof n'envoie pas forcément un mail avec les modalités de rendu! Cette année [2021-2022] c'était marqué sur moodle et elle l'a dit en cours. Ne négligez pas la réalisation des tests, ça représente une grosse partie de l'évaluation. Pas besoin d'être très fort en graphes pour réussir le BE, le plus difficile, c'est de comprendre l'organisation du code qu'on vous donne (vive Java).

Culture et compétences numériques 2

C'est facile

INFORMATIQUE ET RÉSEAUX

Signal aléatoire

C'est des probas avec du signal, si ils ont pas changé par rapport à cette année [2021-2022], le cours est rushé en deux semaines et l'exam est le premier du semestre. Beaucoup de calculs d'intégrales et de probas de base (calcul d'une espérance), ne vous laissez pas impressionner par la physique qui fait peur, la matière est très mathématique. En 2024-2025 on a eu le même sujet que les IMACS avaient eu donc hésitez pas à leur demander leur sujet une fois qu'ils ont passé l'examen (normalement avant vous).

Filtrage Numérique

En cours on ne comprend pas vraiment ce qu'on fait ni à quoi ça sert, mais en TD tout s'éclaire et la matière n'est pas bien compliquée. L'exam peut être assez déconcertant parce qu'il est beaucoup plus appliqué que les TDs (avec des questions qui font appel au bon sens et à la culture G plutôt qu'au cours) mais il est tout à fait faisable en ayant fait quelques annales avant.

Télécom

La première partie récapitule pas mal la télécom. Il vaut mieux aller aux CM car comme elle le dit il y a des infos qu'elle ne donne que là. Certains TD font peur mais normalement le QCM n'est pas dessus.

<https://www.youtube.com/embed/TPS-eFmdepw>

BE Réseaux

Pas de difficulté majeure bosser en dehors et des TP un peu et ça ira

Programmation système et multithreading



(ça fait 3 ans que c'est le même exam après donc si vous le chopez vous pourrez peut-être survivre) et les fiches de cyber-aides sont bien aussi.

AIPS

Les 2 TP ne suffisent pas en général pour finir le code, et le qcm posent quelques questions sur la partie de Nicolas Van-Wambeke et pas mal sur la programmation socket de Chassot (notamment un pseudo-code de client serveur à faire pour nous qui était à écrire dans le qcm). Si vous trouvez les questions ce sera sans doute les mêmes, il faut vraiment bien connaître le poly et avoir bien compris le cours / TP.

Petite vidéo sur le fonctionnement de la [pile et de la mémoire](#)

Réussir son S2 de 3MIC-MA Starter Pack™

Les bases à avoir avant d'arriver en cours:

-

Les liens ressources

- [Drive GAMMA](#)

Conseils plus pratiques

Général:

Matière 1:

Matière 2:

Matière 3: