Numérique et sciences informatiques
Tle

Le préambule du programme sera conservé.

Le programme de terminale sera un prolongement de celui de 1re et on conserve les thèmes.

Il correspond à un enseignement de 100 h, hors projet. La part du projet est plus importante en terminale avec environ 1/3 du temps. Si en 1re les élèves ne peuvent que se consacrer à des projets d'ampleur réduite, en terminale, on peut imaginer qu'un élève se consacrer à un seul projet durant l'année, d'autant que ce projet peut être l'objet de l'épreuve orale terminale.

REPRÉSENTATION DES DONNÉES

▪ **Structures de données abstraites :** interface (description, méthodes), implémentation.

Exemple cité : file de priorité (pas explicite dans le programme).

▪ **Vocabulaire de la programmation objet :** classe, attribut, méthode. Pas d'héritage ni de polymorphisme.

▪ Structures au programme : liste, files, dictionnaires, arbres, graphes (vus aussi en algorithmique).

TRAITEMENT DE DONNÉES EN TABLES

En 1re : Python, CSV, tri… en Tle : les bases de données !

▪ **Modèle relationnel et vocabulaire associé**

▪ ▪ identifier les services rendus : sécurisation, accès. Pas de programmation. Pas de théorie.

▪ **Requêtes :** SQL. Mise à jour de tables, interrogation. Voir programme de Prépa.

Type d'exercice : une table étant donnée, on demande de la corriger.

INTERFACE HOMME-MACHINE ET WEB

Pas tellement plus qu'en 1re car cela se prête plutôt aux projets, mais architecture en trois parties : client/serveur/d=serveur de base de données.

ARCHITECTURES MATÉRIELLES ET SYSTÈMES D'EXPLOITATION

▪ **Composants intégrés sur puce**

▪ **Gestion de processus :** gestion, ordonnancement, interblocage. But : observer un processus

▪ **Protocole de routage :** débranché (table de routage), type de routage. Mettre en évidence que des connaissances locales suffisent.

▪ **Sécurisation et communication :** pas de maths (arithmétique) compliquées car les élèves n'ont pas choisi l'option Maths (expertes) : chiffrement symétrique et asymétrique. Pas de RSA.

Exemple du https : deux chiffrements en même temps…

LANGAGES ET PROGRAMMATION

Poursuivre la 1re pour ce qui concerne les pratiques générales : utilisation de bibliothèques, spécifications, documentation d'un programme, chasse aux bugs…

▪ **Nouveauté :** la récursivité

▪ **Modularité :** utiliser des API (?), créer des modules simples.

▪ **Notion de compilation :** un programme en entrée d'un programme.

▪ **Calculabilité et décidabilité :** faire comprendre qu'il existe différents types de calculabilité. Pas de lambda-calcul. Faire comprendre la thèse de Church. Notion d'indécidabilité : on peut parler d'indécidabilité algorithmique, toutes les fonctions ne sont pas calculables.

Exemple en python : Le problème de l'arrêt d'un programme est indécidable, on obtient une contradiction facilement.

Mais pas de théorie !

ALGORITHMIQUE

▪ **Algorithmique sur arbre :** taille, recherches dans un arbre

▪ **Graphes :** parcourir un graphe, détection de cycle (?), recherche de chemin, algorithme de Dijkstra.

▪ ▪ **Méthode diviser pour régner :** rotation bitmap à coût constant, tri fusion.

Programmation dynamique : alignement de séquences, rendu de monnaie

▪ **Algorithmique sur texte :** algorithme de Boyer-Moore (recherche d'un mot dans un texte), avec un coût sous-linéaire, sans balayage du texte. Pas d'étude de coût.