PROGRAMME ET INSTRUCTIONS *OFFICIELLES*

POUR L'ENSEIGNEMENT DE

L'INFORMATIQUE

Première et deuxième années des classes préparatoires filières : MP, PSI et TSI

PREAMBULE

Les technologies de l'information et les sciences du numérique sont aujourd'hui utilisées dans les différents secteurs d'activités et spécialement en ingénierie. Aussi leur intégration dans le cursus des Classes Préparatoires aux Grandes Ecoles (CPGE) est une nécessité pour préparer tous les élèves à leurs futurs métiers.

Ces technologies de l'information étant en perpétuelle évolution, le programme d'enseignement relatif à ces disciplines est amené aussi à être modifié continuellement afin de s'adapter à cette évolution permanente.

Ainsi, Le Ministère de l'Education Nationale entreprend régulièrement d'importants efforts pour réviser les programmes d'informatique dans les classes préparatoires marocaines. A cet effet, ce présent guide a été élaboré pour permettre de :

- définir la nature et les caractéristiques de l'informatique en tant que discipline d'enseignement en CPGE;
- délimiter le cadre et la vision du programme d'informatique en C.P.G.E ;
- indiquer les compétences à développer chez les apprenants;
- fixer les finalités et les objectifs de chaque partie du programme;

- préciser les approches pédagogiques servant de référence pour préparer les activités d'apprentissage relatives à la discipline informatique ;
- présenter le programme ainsi que la progression qui lui est attachée;
- proposer des exemples illustratifs des notions fondamentales de ce programme
- Suggérer quelques méthodes et moyens permettant à l'enseignant de planifier, d'animer et d'évaluer les apprentissages des élèves.
- Conseiller des exercices et des exemples d'applications relatifs aux différents éléments de ce programme

SOMMAIRE

-

A-Contexte de la nouvelle réforme de l'informatique en C.P.G.E

Le programme d'informatique des classes préparatoire aux grandes écoles d'ingénieurs tient compte des évolutions de l'enseignement des sciences et technologies du numérique dans l'enseignement secondaire et les écoles d'ingénieurs.

Il vise à doter les élèves des *CPGE* de méthodes, d'outils et de concepts nécessaires à la résolution informatique d'un problème donné. Ce problème peut s'appuyer sur les disciplines des mathématiques, de la physique, de la chimie ou des sciences industrielles.

Cette résolution informatique est selon les cas soit la conception d'un algorithme qui peut être traduit en programme informatique, soit l'utilisation d'un logiciel de simulation numérique ou encore la gestion d'une base de données.

L'élève devra savoir appréhender un problème simple du monde naturel pour le traduire, l'implémenter en une solution informatique. Il devra se familiariser avec différents concepts comme la faisabilité d'une solution donnée, la précision des calculs numériques, l'efficacité d'un programme, etc.

Les élèves devront suivre aussi bien une formation à caractère théorique qu'une formation à caractère pratique.

Ce sont ces 3 aspects, à savoir l'algorithmique et programmation, la simulation numérique et la gestion des bases de données qui constituent le fondement de ce programme d'enseignement d'informatique en *CPGE dans les filières MP, PSI et TSI*.

B-Capacités et objectifs requises

Ce programme d'informatique, n'a pas vocation seulement d'apprendre aux élèves les concepts liés au domaine des technologies de l'information mais aussi les aider à adopter des démarches scientifiques et des approches appuyées sur des raisonnement rigoureux conduisant aux solutions des problèmes posés.

Ainsi, l'enseignement de l'informatique doit permettre de développer chez les élèves les capacités et objectifs suivants :

- comprendre, analyser et modéliser un problème ou une situation pouvant être traité par l'informatique,

- adopter une démarche de travail (définir le cahier de charge (les données, les résultats attendus, les traitements à effectuer, …),
- comprendre un algorithme et expliquer son déroulement et son rôle,
- choisir les structures pouvant représenter les données dans la mémoire r
- découper le problème en plusieurs parties (modules),
- concevoir l'algorithme de résolution relatif à chaque module du problème en spécifiant de façon rigoureuse et structurée l'ensemble des étapes,
- mettre en pratique l'algorithme en le traduisant en programme à l'aide d'un langage informatique,
- tester, vérifier et éventuellement corriger les algorithmes et les programmes résultants,
- valider la solution algorithmique par des jeux d'essais,
- exploiter les fonctionnalités et les performances qu'offre un logiciel de simulation numérique,
- gérer et interroger une base de données,
- critiquer, modifier, les données manipulées,
- spécifier les modules mis en œuvre,
- évaluer la complexité des différentes fonctions et des divers programmes réalisés.

La résolution des problèmes par une approche algorithmique procure aux élèves une rigueur, une capacité d'abstraction, un raisonnement et une logique, qui sont les éléments essentiels pour toute résolution d'un problème scientifique.

Ces capacités acquises, permettront leur intégration dans les autres enseignements scientifiques ainsi que dans l'élaboration des travaux d'initiative personnelle encadrée (*TIPE*).

C- Structure, objectifs et méthodologie du programme

C-1 Structure et organisation du programme

Le programme de l'informatique des deux années des *CPGE* scientifiques, est organisé en 5 parties, comme suit :

- Partie I : Généralités et algorithmique de base

- Partie II: Algorithmique et programmation

- Partie III : Ingénierie numérique et simulation

- Partie IV : Introduction aux bases de données relationnelles

- Partie V : Algorithmique avancé et programmation

Les trois premières parties (partie I, II et III) seront dispensées durant la première année alors que les deux dernières parties (partie IV et V) concerneront la deuxième année.

C-2/ Objectifs fixés

L'enseignement de l'informatique aux *CPGE* vise plusieurs objectifs permettant d'inculper aux élèves des outils et des méthodes nécessitant l'application des règles de logique absolue et favorisant la réflexion, l'analyse et la synthèse. Ces capacités sont nécessaires pour résoudre tout problème d'ordre scientifique.

Ainsi ce nouveau programme d'informatique, n'a pas vocation seulement d'apprendre aux élèves les concepts liés au domaine des technologies de l'information mais aussi les aider à adopter des démarches scientifiques et des approches appuyées sur des raisonnement rigoureux conduisant aux solutions des problèmes posés.

Les principaux objectifs de cette nouvelle réforme de l'enseignement de l'informatique aux *CPGE*, peuvent se résumer aux points suivants :

- apprentissage d'un ensemble de concepts de base de l'algorithmique,
- résolution de problème par approche algorithmique,
- conception rigoureuse d'algorithme,
- choix des représentations appropriées des données,
- familiarisation avec la syntaxe élémentaire et les règles de base d'un langage de programmation,
- mise en œuvre d'un programme informatique (édition, exécution, test),
- pratique de la programmation,
- utilisation d'un logiciel de simulation numérique,
- exploitation des résultats de calculs numériques,
- initiation à l'utilisation d'un SGBD,
- manipulation de données organisées dans une base de données,
- évaluation de l'efficacité algorithmique (mesure de la complexité),
- maitrise de l'approche récursive d'un algorithme.

C.3/ Méthodologie

* Il est recommandé de mettre l'accent sur le raisonnement algorithmique beaucoup plus que sur la syntaxe du langage de programmation.

* Il est conseillé de donner des exemples et de proposer des exercices inspirés des autres disciplines scientifiques (mathématiques, physique, sciences de l'industrielle et chimie).

* Il est nécessaire de transcrire les algorithmes vus pendant les séances de cours en langage de programmation.

* Il est souhaitable d'intégrer des éléments de la discipline informatique dans l'élaboration des travaux d'initiative personnelle encadrée(*TIPE*).

* Il n’est pas demandé d’aborder les aspects théoriques qui relèvent des autres disciplines scientifiques, mais de mettre en œuvre les algorithmes permettant l’analyse des résultats surtout concernant l’utilisation du logiciel de simulation.

* Il est à noter que le langage de programmation, le logiciel de simulation numérique ainsi que le système de gestion de bases de données (*SGBD*) qui seront utilisés dans le cadre de ce cours feront l’objet d’une note de service émanant de l’autorité gouvernementale chargée des classes préparatoires aux grandes écoles.

D- Programme de la première année (MPSI, PCSI, TSI)

Le programme de la première année est divisé en 3 parties (I, II et III)

D.I Partie I : Généralités et algorithmique de base

La partie **I** du programme sera enseignée durant le premier trimestre de la 1ère année. Le but de ce paragraphe est d'initier les élèves à la conception d'algorithmes élémentaires manipulant des variables de types simples, des expressions et des structures de contrôle, puis convertir ces algorithmes en programmes à l'aide d'un langage de programmation.

D.I.a/Capacités visées

- identifier l'ensemble des ressources physiques constituant le système informatique,
- utiliser les principales commandes d'un système d'exploitation,
- comprendre la représentation des données numériques dans la mémoire et les limitations (en valeur et en précision) de leur représentation,
- expliquer ce que fait un algorithme élémentaire,
- décomposer une tâche complexe en tâches élémentaires
- concevoir un algorithme répondant à un problème simple,
- savoir adopter l'approche de la programmation descendante et la programmation structurée,
- se familiariser avec la syntaxe de base d'un langage de programmation,
- traduire un algorithme en programme écrit avec un langage informatique,
- vérifier la terminaison d'une boucle,
- tester un programme en l'éditant et en l'exécutant dans un environnement intégré d'un langage de programmation,
- appeler quelques fonctions élémentaires du langage de programmation.

D.I.b/ Outils utilisés

Les vacations de cours théoriques seront alternées avec de séances de travaux dirigés et de travaux pratiques.

Durant les travaux dirigés, les élèves apprennent à concevoir des algorithmes de résolution de problèmes élémentaires et à écrire les programmes correspondants.

Ces programmes seront édités et testés au cours des travaux pratiques prévus

D.I.c/ Contenu

D.I.c.1/ Environnement matériel et logiciel d'un système informatique

Le but de ce paragraphe est de présenter succinctement l'ensemble des ressources physiques constituant le système informatique ainsi que l'ensemble des moyens qui permettent d'utiliser l'ordinateur. On abordera aussila représentation des données numériques sous forme binaire dans la mémoire.

a/ Architecture simplifiée d'un ordinateur

***Composants de base**

- ordinateur personnel, tablette, *PDA*,
- différencier les différents composants constituant un ordinateur (mémoire vive, mémoire de masse, unité centrale, périphériques d'entrée-sortie, ports de communication…)

*** Mémoire centrale et représentation des données**

- rôle, caractéristiques, capacité (notion de bit, de mot mémoire, d'adresse mémoire, …).

* **Représentation de données dans la mémoire**

- types de données (nombres ou caractères),
- système binaire,
- représentation des nombres entiers en mémoire (nombre positif ou négatif, bit de signe, nombre non signé),
- représentation des nombres réels en mémoire (mantisse et exposant, définition de l'écriture en virgule flottante normalisée)
- limites et dépassement de la mémoire (illustration par des exemples simple (division par zéro,…)),
- codage et représentation des caractères (code *ASCII*, *UTF8*, *UTF16*, *Unicode*…),
- étude des mécanismes de dépassement de capacité (les "overflow"), les erreurs d'arrondis, les divisions par zéro.

* **Processeur**

- caractéristiques (vitesse, fréquence de l'horloge, …),
- fonctions (opérations arithmétiques et logiques).

*** Périphériques**

- périphériques d'entrée-sortie, ports de communication, Mémoires auxiliaires,…).

b/ Système d'exploitation d'un ordinateur

- définition et exemples de systèmes d'exploitation (*Windows* et *Linux*),
- fonctions principales d'un système d'exploitation (vérification des ressources, gestion de fichiers, gestion de dossiers, …),
- exemples d'utilisation de quelques commandes usuelles (création de dossiers, copie de fichiers, …),
- mise en œuvre d'un environnement de développement.

D.I.c.2/Algorithmique de base

Le but de ce paragraphe est d'initier les élèves à la conception d'algorithmes élémentaires manipulant des variables de types simples, des expressions et des structures de contrôle, puis convertir ces algorithmes en programmes à l'aide d'un langage de programmation

a/Définitions

* algorithme et programme,
*langage de programmation,
*interpréteur et compilateur.

b/Eléments de base d'un algorithme

*variables.
*types simples (entier, flottant, caractère),
*affectation,
*entrées / sorties standards et fonctions de la bibliothèque.

c/Opérateurs et les expressions

*les opérateurs arithmétiques et logiques,
* les expressions.

d/Structures de contrôle

*les instructions de choix (la sélection logique),
*les instructions d'itération ou de répétition (les boucles).

e/ Démarche d'analyse descendante

*principe de la démarche,
*affinements successifs,
*exemples.

f) Modélisation.

*analyser une situation,
*spécifier,
*modéliser.

g) Vérification du code

*invariants de boucle,
*tests des segments itératifs,
*estimation de la complexité d'un calcul.

D.I.d/ Exemples d'exercices et d'applications pratiques

La liste suivante énumère quelques exemples d'algorithmes simples pouvant être proposés en termes d'exemples ou d'exercices durant cette première partie de l'enseignement :

- Résolution d'une équation du 1er ou du 2ème degré
- Vérification de la validité d'une date (années bissextile).

- Détermination de certaines propriétés d'un nombre entier (premier, parfait, *Amstrong*, …).
- Calcul de *PGCD* et du *PPMC* de 2 nombre entiers.
- Décomposition binaire d'un entier.
- Evaluation de la somme des N premiers nombres entiers positifs.
- Calcul du factoriel d'un nombre en utilisant une méthode itérative.
- …

D.II Partie II : Algorithmique et programmation

La partie**II** du programme sera enseignée durant le deuxième trimestre de la 1ère année.Le but de cette partie est d'apprendre aux élèves la programmation modulaire ainsi que la représentation des données dans des structures des données telles que les tableaux, les chaînes de caractères, les listes, les piles, …) et leurs utilisations (tri, recherche, …). Dans cette partie, on s'intéressera aussi à la lecture et à l'écriture dans un fichier de données.

D.II.a/Capacités visées en algorithmique

- découper et organiser le problème en plusieurs parties (les modules),
- définir les prototypes des fonctions à utiliser (paramètres, type de retour)
- choisir les structures pouvant représenter les données et les résultats du problème,
- maitriser les algorithmes de tri standard et de la recherche dichotomique
- manipuler une chaîne de caractères,
- gérer une liste de données (ajouter, modifier, rechercher ou supprimer des éléments),
- organiser les éléments sous forme d'une pile afin de leur exploitation,
- lire et écrire des données à partir d'un fichier.

D.II.b/ Capacités visées en programmation

Traduire un algorithme dans un langage de programmation non spécifiquement liés au système d'exploitation, libre et portable, dynamique, extensible qui permet une approche modulaire et orientée objet de la programmation.

Savoir se servir de la documentation du langage *choisi* en ligne,

- savoir documenter ses propres programmes,
- choisir les types et structures de données optimales pour un problème donné,
- expliciter la signature, le prototype d'une fonction,
- mettre en œuvre des séquences de tests afin de tester une fonctionnalité, un programme spécifique.

D.II.c/Contenu en programmation

* présentation des différents types d'objets,
* objets simples : entier, flottant, complexe, booléen et chaines de caractères,
* les opérateurs,
* les collections d'objets (les listes, les tuples, les dictionnaires, les ensembles),
* la logique d'un programme (les opérations conditionnelles, les boucles *for* et *while*),
* les fonctions (les instructions break et continue, variable locale, variable globale),
* les modules de base du langage choisi.

D.II.d/Outils utilisés

La programmation modulaire, les structures des données ainsi que les principes des différents tris seront expliqués pendant les séances de cours. Les différents algorithmes de tri et de recherche seront conçus en travaux dirigés.

L'implémentation et la vérification de ces différents algorithmes seront effectuées en travaux pratiques à l'aide du langage.

Les élèves devront utiliser un environnement de développement

Un environnement scientifique doit être présenté et ceci dès la première année. Cela peut-être un logiciel de calcul scientifique libre et à large utilisation ou une bibliothèque utiliser avec le langage de programmation choisi.

Attention il n'est pas demandé une étude approfondie de l'ensemble de ces outils.

Des textes réglementaires pourront dans le futur préciser les différents choix d'outils et d'environnement de développement.

D.II.e/Contenu

D.II.e.1/Programmation modulaire

a. Définition et paramètres de fonction,
b. Variable locale et variable globale,

D.II.e.2/Structures de données et leurs utilisations

a. Tableaux à une dimension (représentation, accès aux éléments),
b. Chaînes de caractères (longueur, accès à un caractère),
c. Tris et recherche dichotomique dans un tableau (tri par sélection, tri par insertion, tri à bulles, …),
d. Tableaux à 2 dimensions (opérations sur les matrices),
e. Listes (définition, ajout, suppression, recherche dans une liste),
f. Piles et files.

D.II.e.3/Fichiers de données

a. Accès à un fichier (chemin et nom physique de fichier),

b. Lecture des éléments d'un fichier à partir d'un programme,

c. Ecriture dans un fichier à partir d'un programme.

D.II.f/Exemples d'exercices et d'applications pratiques

Les exemples ci-dessous peuvent être suggérés en exercices :

- Gestion d'une une liste d'éléments et son tri (ajout, modification, suppression, recherche, …).
- Calcul matriciel (somme, produit, déterminant, transposée).
- Utilisation des algorithmes de cryptage élémentaires (*César*, *Vigenère*)
- Cryptage d'une chaîne de caractères puis d'un fichier texte.
- Implémentation de l'algorithme du codage de *Hamming* et de *Huffman*.
- Vérification de la syntaxe d'une expression mathématique simple (nombre de parenthèses par exemples) en utilisant la structure **pile.**

D.III Partie III : Ingénierie numérique et simulation

Le but de cette partie est d'introduire les techniques de base pour l'utilisation des algorithmes numériques. On ne s'intéressera qu'à la mise en œuvre de ces algorithmes et non pas aux aspects théoriques. Par contre il faudra comparer les solutions obtenues d'une manière numérique avec celle issue d'un calcul analytique

D.III.a/Compétences visées

À la fin de cette partie, l'élève doit être capable de :

- Donner une solution numérique d'un problème difficilement ou impossible à réaliser avec une solution purement analytique
- Mettre en œuvre par programmation un problème à caractère scientifique en prenant en compte aussi bien les données en entrées du programme que les résultats
- Savoir utiliser les bibliothèques de calcul numérique standard pour résoudre un problème scientifique
- Savoir utiliser les bibliothèques standards pour afficher les résultats sous forme graphique
- Savoir prendre en compte les erreurs de calcul ou erreurs d'arrondi.
- Savoir déterminer la complexité d'un calcul en vue d'en déterminer si le calcul est réalisable dans un temps acceptable.

D.III.b/Outils utilisés

Après chaque séance théorique, un ensemble d'exercices et de travaux pratiques seront proposés aux élèves pour illustrer et vérifier les notions vuesen cours.

D.III.c/Contenu

D.III.c.1/Présentation des bibliothèques

* Programmation de quelques fonctions.
* Calcul manipulant les tableaux et calcul matriciel
* Gestion de la documentation en ligne des fonctionnalités de ces bibliothèques.

D.III.c.2/Méthodes linéaires à une dimension

* Résolution d'équation algébrique
* Méthode de dichotomie
* Méthode de *Newton*

D.III.c.3/Problèmes dynamiques à une dimension

* Résolution approchée d'une équation différentielle
* Méthode d'*Euler*

D.III.c.4/ Problèmes discrets multidimensionnels linéaires

* Résolution d'un système linéaire inversible à l'aide de la méthode de *Gauss*
* Calcul de complexité pour déterminer un temps d'exécution afin de savoir si le calcul est envisageable.

D.III.d/ Exemples d'exercices et d'applications pratiques

- Manipulations de tableaux et de matrices (extraire un sous tableaux, coupe d'un tableau ou d'une matrice, indexation par masque, copie de tableau ou de matrice, opérations simples ou complexes manipulant des tableaux ou des matrices, etc)
- Exemple : équation de la chaleur (diffusion thermique) à une dimension discrétisation en différences finies, convergence, implémentation matricielle, optimisation avec la formule de *Taylor*, etc)
- Intégration d'équations différentielles à l'aide d'une bibliothèque ou d'un logiciel de calcul scientifique
- Exemple : gestion d'un système dynamique, le pendule simple.

E. Programme de la deuxième année (MP, PSI, TSI)

Le programme de la deuxième année est divisé en **2** parties (partie **IV** et partie **V**)

E.I Partie IV: Introduction aux bases de données

La partie **IV** du programme sera enseignée durant la première période de la 2ème année .Cette partie vise à initier les élèves à la technologie des bases de données.

E.I.a/ Capacités visées

* Mettre en œuvre et gérer une base de données relationnelle;
* Effectuer des requêtes SQL sur une base de données ;
* Découvrir et pratiquer les principales fonctions d'un SGBD relationnel ;
* Acquérir une expérience pratique dans la gestion d'une base de données.

E.I.b/ Outils utilisés

Cette partie sera dispensée sous forme de présentation des concepts élémentaires des bases de données. Chaque concept doit être illustré par un ou plusieurs exemples simples. Ces concepts seront mis en œuvre dans des séances de travaux pratiques.

Ces travaux pratiques seront réalisés avec un SGBD qui présente une interface graphique conviviale permettant l'utilisation des différentes manipulations sur les bases de données (création de bases et de tables, alimentation, modification, suppression et interrogation des données)

Il est à rappelé que le système de gestion de bases de données (SGBD) à utiliser fera l'objet d'une note de service de l'autorité gouvernementale chargée des classes préparatoires aux grandes écoles (voir page 7)

E.I.c/ Contenu

E.I.c.1/ Généralités

* Notion de base de données;
* Modèle de base de données (hiérarchique, relationnel);
* Système de gestion de base de données (SGBD);
* Machine serveur et client;

E.I.c.2/ Modèle relationnel

* Présentation;
* Concepts élémentaires;
* Notions de base (table, relation; attribut; *clés; domaine; …) ;*
**Schéma de relation;*

E.I.c.3/ Algèbre relationnelle

* Introduction;
* Opérateurs unaires (sélection, projection);
* Opérateurs binaires ensemblistes (union, intersection, différence);
* Opérateurs n-aires(Produit cartésien, jointure, division cartésienne);
* Fonctions d'agrégation : min, max, somme, moyenne,...

E.I.c.4/ Langage SQL

* Présentation et notion de requête;
* Description de données (création, modification suppression de tables, d'attributs, de vues, …);
* Manipulation de données (Insertion, modification suppression de n-uplets);
* Interrogation d'une base de données (commande **SELECT**);

E.I.d/ Exemples d'exercices et d'applications pratiques

* Lancer l'environnement d'un SGBD offrant une interface graphique et créer une base de données simple contenant quelques tables (2 ou 3) ;
* Définir la structure d'une table (champs, types,..) ;
* Alimenter les tables de la base ;
* Interroger la base à l'aide des requêtes ;
* Ecrire des requêtes SQL en utilisant l'algèbre relationnelle à partir d'énoncés en langage courant ;
* Accéder à une base de données déjà existante et manipuler ses données ;
* Utiliser les jointures symétriques simples (JOIN … ON …=...) pour effectuer des requêtes croisées dans une base de données constituée de plusieurs tables.
* Présenter à l'aide d'un exemple simple le concept de client-serveur en précisant les rôles respectifs des machines client, serveur, et éventuellement serveur de données (architecture trois-tiers)

Remarque : Les exemples d'illustration de bases de données seront de préférence choisis au sein des autres disciplines scientifiques étudiées.

E.II Partie V: Algorithmique et programmation 2

La partie **V** du programme est la dernière partie de cet enseignement, elle sera dispensée durant la deuxième période de la 2ème année

E.II.a/Compétences visées

Le but de cette partie est d'introduire certaines notions avancées d'algorithmique et quelques compléments qui n'ont pas été abordés en 1ère année. Ainsi à la fin de cette partie, l'élève doit être capable de :

* Evaluer la «**qualité**» d'un algorithme en mesurant sa complexité ;
* Résoudre des problèmes en utilisant les algorithmes récursifs ;
* Comprendre les avantages et les inconvénients des approches itérative et récursive ;
* Représenter des données dans un arbre binaire et le parcourir;
* Comprendre les mécanismes de base de la programmation objet (classe, attributs, instance, méthodes statiques, méthodes à instances, …) ;
* Utiliser les expressions régulières

E.II.b/ Outils utilisés

Après chaque séance théorique, un ensemble d'exercices et de travaux pratiques seront proposés aux élèves pour illustrer et vérifier les notions vues en cours.

Remarque: Le langage de programmation utilisé est le même que celui adopté en 1ère année.

E.II.c/ Contenu

E.II.c.1/ Algorithmes de tri

* Rappels des algorithmes de tri standard (sélection, insertion, bulles) ;
* Algorithmes de tri rapides (Quick Sort, Merge Sort) ;

E.II.c.2/ Initiation à la complexité algorithmique

* Notion de complexité algorithmique (Complexité en temps et en espace) ;
* Algorithmes de complexité (Algorithme constant, logarithmique, linéaire) ;
* Comparaison de la complexité temporelle d'un algorithme de tri standard et un algorithme de tri rapide ;

E.II.c.3/ Récursivité

* Principe de la récursivité simple ;
* Exemples d'utilisation de la récursivité simple ;
* Occupation de la mémoire (Pile) et problème de saturation ;
* Terminaison d'une fonction récursive ;
* Récursivité et Itération (comparaison);

E.II.c.4/ Initiation aux algorithmes des arbres binaires

* Définition d'un arbre binaire (notion de nœud, racine, feuille) ;
* Parcours d'un arbre binaire ;
* Exemple du tri Maximier (tri en utilisant un arbre binaire) ;

E.II.c.5/ Introduction à la programmation orientée objet

* Notion d'objet, de classe, d'instance, d'attributs
* Constructeurs ;
* Méthodes (statiques et d'instances);
* Instanciation, accès aux attributs, appel aux méthodes;

E.II.c.6/ Expressions régulières

* Introduction (définition d'une expression régulière, rôle,
* Symboles dans les Expressions Régulières (lettres, *, +,?,)* Syntaxe des expressions régulières

E.II.d/ Exemples d'exercices et d'applications pratiques

* Evaluer la complexité d'un algorithme simple de complexité constante (recherche séquentielle d'un élément dans une liste) ;
* Implémenter un algorithme de tri standard (tri par sélection) et un algorithme de tri rapide (Quick Sort) puis déterminer et comparer leurs complexités dans le pire et le meilleur des cas ;
* Développer des algorithmes récursifs (factorielle, puissances entières, suites récurrentes (Fibonacci,…), pgcd,…) ;
* Représenter des données sous forme d'un arbre binaire exemple : arbre généalogique) et rechercher une information en parcourant l'arbre ;
* Comprendre le principe du tri par arbre (tri Maximier);
* Découvrir la programmation orientée objet au travers d'applications simples
* Application aux expressions régulières (la validité d'une adresse e-mail,)