MINISTÈRE
DE L'ÉDUCATION
NATIONALE,
DE LA JEUNESSE
ET DES SPORTS

Liberté
Égalité
Fraternité

Concours externe de l'agrégation du second degré

Section informatique

Programme de la session 2022

L'agrégation d'informatique a été créée par un arrêté publié au JO du 13 juin 2021[1] qui définit également ses épreuves. Comme indiqué,

Les épreuves 1° et 2° d'admissibilité portent sur les programmes d'enseignement de la spécialité « numérique et sciences informatiques » (NSI) du cycle terminal de la voie générale du lycée, ceux des classes préparatoires scientifiques aux grandes écoles

« Mathématiques, physique, ingénierie et informatique » (MP2I) et « mathématiques, physique, informatique » (MPI), auxquels s'ajoute un programme complémentaire publié sur le site internet du ministère chargé de l'éducation nationale.

L'épreuve 3° d'admissibilité porte sur les programmes mentionnés ci-dessus aux- quels s'ajoute un programme complémentaire spécifique à chacun des sujets au choix (étude de cas informatique ou fondements de l'informatique). Ce programme complémentaire fait également l'objet d'une publication sur le site internet précité. Pour l'ensemble du programme, il est attendu du candidat un recul correspondant au niveau master.

Ce document définit les programmes complémentaires pour ces épreuves d'admissibilité. On rappelle que ce programme est complémentaire et que le programme de l'agrégation d'informa- tique contient en premier lieu les programmes de NSI et d'informatique en MP2I et MPI. Certains éléments de ces programmes sont rappelés dans la présentation qui suit, afin de donner au présent programme complémentaire une forme cohérente.

Pour des raisons de temps, quelques points de ce programme restent à compléter et une version définitive sera disponible au même endroit avant septembre 2021.

Conformément aux programmes en vigueur, la connaissance des langages OCaml, C et Python est attendue, ainsi que la connaissance du langage de requêtes SQL. Les programmes des classes MP2I/MPI fixent dans leur annexe les éléments des langages C et OCaml dont la connaissance est exigible.

Une fiche similaire sera fournie pour Python.

1 Épreuves 1 et 2 d'admissibilité

Algorithmique

- Pas de programme complémentaire.

Logique

- Pas de programme complémentaire.

Architecture

- Circuits combinatoires/séquentiels, machine de Mealy, machine de Moore.
- Description et fonctionnement d'une machine de von Neuman (introduction à la programmation assembleur : instructions de calcul, instructions de branchement, accès à la mémoire et aux registres, instructions système).
- Exécution d'un appel de fonction, concept de pile.

1 https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000043648279

- Hiérarchie mémoire.
- Typologie des machines parallèles (classification de Flynn, classification de Raina, machines multi-coeurs, supercalculateurs).
- Représentation des flottants. Problèmes de précision des calculs flottants et de dépassement de capacité. Notion de mode d'arrondi.

Bases de données

- Requêtes conjonctives et calcul conjonctif.
- Opérateurs de l'algèbre relationnelle et leurs propriétés : application à l'optimisation derequêtes.

Calculabilité, complexité

- Modèle de calcul. Machines de Turing : définition, principales variantes (ruban biinfini vs infini, machine à plusieurs rubans). La machine de Turing est le modèle de calcul retenu pour l'étude des notions qui suivent.
- Calculabilité : universalité, décidabilité, indécidabilité. Problème de l'arrêt.
- Complexité : complexité en temps et en espace, classe P. Acceptation par certificat, classeNP. Réduction polynomiale. NP-complétude. Théorème de Cook.

La notion de machine de Turing non déterministe n'est pas exigible aux épreuves 1, 2 et 3.a de l'écrit, ni aux épreuves orales.

Chaîne de compilation

- Analyse lexicale, analyse syntaxique (principes de l'analyse descendante).
- Analyse sémantique élémentaire (arbre de syntaxe abstraite, environnement, analyse deportée, typage).
- Génération de code vers une machine à pile.

Informatique et société

- Points clés du RGPD.
- Notions de Données d'Intérêt Général (DIG).
- Impact environnemental du numérique. Notion d'écoconception.
- Points clés de la loi n° 2016-1321 du 7 octobre 2016 pour une République numérique.
- Enjeux d'éthique du numérique et valeurs sous-jacentes.

Intelligence artificielle

- Apprentissage machine, mesures de similarité.
- Données d'entraînement et données de test, choix des descripteurs.
- Enjeux d'éthique (biais d'apprentissage, transparence).

Génie logiciel

- Production de logiciel : cycle de vie, cycle de développement, agilité

Programmation

- Programmation objet : objets, classes, héritage, polymorphisme de sous-typage.
- Programmation fonctionnelle : ordre supérieur, structures immuables, polymorphisme paramétrique.

Réseaux

- A compléter

Systèmes d'exploitation

- Abstractions fournies par le système : de l'initialisation à l'accès au matériel, répartition équitable des ressources.
- Liens entre système d'exploitation et applications : adressage physique et virtuel, notion de pagination, MMU, interruptions, appels systèmes, gestion des processus, gestion du temps (ticks et implémentation du temps partagé).
- Isolation et interaction entre les processus : espace mémoire d'une application, communiation entre applications (*pipe*, mmap).
- Concurrence : modèles de cohérence (forte, faible, PRAM et au relâchement) et d'équité. Construction des mutex et sémaphores à partir des instructions atomiques test_and_set. Schéma lecteurs rédacteurs.

2 Épreuves 3.a – étude de cas informatique

Génie logiciel

- Analyse : identification et spécification des besoins fonctionnels, définition des scénarios d'utilisation (séquence d'interactions).
- Conception : architecture logicielle (en couches), conception modulaire et approche parpatron de conception (*Pattern*). Diagramme de classes / de tables.
- Maintenance : test unitaire, *Refactoring*.

Une connaissance exhaustive des patrons de conception n'est pas exigible à cette épreuve.

Programmation web

- HTML / CSS : pas de programme complémentaire.
- JavaScript : Principaux objets du DOM, requêtes sur le DOM, modifications du DOM, évènements (mode de propagation et réaction aux évènements), requêtes HTTP (envoi etréception en HTML et en JavaScript), programmation asynchrone (async / await).
- Une fiche fixant les éléments du langages JavaScript dont la connaissance est exigible sera fournie.

Réseaux

- A compléter

Systèmes d'exploitation

- Émulation et virtualisation : types d'hyperviseurs (type 1 vs type 2) et conteneur.
- Virtualisation matérielle et para-virtualisation

3 Épreuves 3.b – fondements de l'informatique

Calculabilité, complexité

- Machines de Turing non déterministes : équivalence avec les machines de Turing du pointde vue de la calculabilité. Définition de la classe NP comme la classe des problèmes résolubles en temps polynomial par une machine de Turing non déterministe, équivalence avec l'acceptation par certificat.
- Définitions et caractérisations des ensembles récursifs, récursivement énumérables.

- Fonctions primitives récursives : définition, schémas primitifs, minimisation bornée. Fonctions récursives : définition, équivalence avec les machines de Turing.
- Lambda-calcul pur comme modèle de calcul : définition, propriétés (dont confluence), stratégies. Équivalence avec les machines de Turing et les fonctions récursives.
- Thèse de Church-Turing.

Logique

- Logique du premier ordre (aspects syntaxiques) : langages, termes, formules, variables libres et variables liées, substitutions, capture de variables.
- Logique du premier ordre (aspects sémantiques) : interprétation d'une formule dans un modèle, validité, satisfiabilité, théories cohérentes, théories complètes, théories décidables, indécidables. Exemples de théories : égalité, arithmétique de Peano.
- Bases de données : calcul relationnel et théorème de Codd.

Fondements de la programmation

- Preuve de programmes : correction, terminaison. Méthodes élémentaires : assertions, préconditions et postconditions, invariants et variants de boucles, logique de Hoare, induction structurelle.
- Sémantique des langages de programmation : sémantiques opérationnelles. Application à un langage impératif restreint (IMP).
- Systèmes de types : types simples. Application à un langage fonctionnel simple (mini-ML sans polymorphisme), sûreté du typage.