

ÉPREUVE ORALE D'INFORMATIQUE FACULTATIVE

Concours	Nb cand.	Moyenne	Ecart type	Note la plus basse	Note la plus haute
A BIO	1396	13,62	3,86	2	20
A ENV	628	14,33	3,70	3	20
A PC BIO	463	14,18	3,72	3	20

Modalités et attentes du jury :

L'épreuve d'oral dure en tout 1 heure : 30 min de préparation et 30 min d'interrogation.

Durant la phase de préparation chaque candidat prépare un des deux exercices qui lui est proposé et se prépare à présenter le projet sélectionné par l'examineur parmi les deux fournis par le candidat au préalable.

Durant l'interrogation orale un temps généralement comparable est alloué à la présentation du projet et à la résolution de l'exercice.

Aussi, les candidats doivent se préparer à pouvoir présenter en 15 min au plus leur résolution de l'exercice et en 10 minutes environ les éléments clés de leur projet afin de laisser un temps de question et discussion autour de ce dernier. L'examineur gère néanmoins le déroulement de l'interrogation dont l'objectif est d'évaluer les compétences des candidats, les exercices proposés servant notamment de trame à cette fin.

Lors de la présentation de son projet le candidat doit faire ressortir clairement les points suivants :

- l'objet d'étude avec ses spécificités et les fonctionnalités associées développées
- les choix de représentations (structure de données) effectués et justifiés
- la structuration du programme (recours à des fonctions pertinentes, modularité et factorisation du code) et les principaux éléments algorithmiquement intéressants
- les exécutions réalisées, les points de blocage et les points d'amélioration envisagés

Par ailleurs, les codes sources fournis dans les dossiers doivent être correctement indentés et commentés à bon escient. Pour chaque fonction, l'objectif, les entrées et les sorties doivent être

précisés. Les portions de codes qui le nécessitent doivent être commentées. De plus, les variables utilisées et fonctions créées doivent porter des noms explicites.

Nous insistons sur le fait que les candidats sont évalués sur la partie projet et sur la partie exercice avec des poids analogues et que par conséquent un candidat présentant honorablement son projet mais étant dans l'incapacité de résoudre l'exercice ne pourra pas prétendre à une bonne note. De même qu'un candidat ayant bien résolu son exercice mais étant dans l'incapacité d'expliquer son projet ou présentant un projet creux ne peut prétendre au mieux qu'à une note moyenne.

Observations :

Remarques générales :

Nous constatons depuis quelques années que le niveau se "moyenne".

Les notions de bases sont mieux connues (variables, fonctions, boucles). Néanmoins, les notions d'entrées et de sorties posent encore problème et la proportion de candidats incapables de répondre à des questions très simples semble assez constante voire peut-être en progression. Par ailleurs, les candidats arrivant à aller au bout de l'exercice semblent moins nombreux.

Partie projet :

Un effort sur les projets a été noté néanmoins comme observé les années précédentes, la qualité moyenne des travaux fournis pour cette partie demeure préoccupante. Nous rappelons que pour les sessions futures cette partie de l'épreuve est conservée.

En effet, le nombre de projet inintéressants ou trop légers d'un point de vue algorithmique demeure important. Il n'est pas rare de voir présentés des projets dont les développements sont à peine équivalents à ce qui est demandé en exercice voire en deçà.

Le jury est conscient de la difficulté de préparation de cette partie de l'épreuve. Néanmoins, si un projet présente des objectifs modestes un soin tout particulier doit être apporté à la qualité du code (factorisation, modularité, structures de données judicieuses, noms explicites pour les variables et fonctions, code propre, bien indenté et correctement documenté).

Il n'est pas toujours évident de proposer des projets dont la partie algorithmique soit particulièrement intéressante mais le travail rendu doit tout du moins faire apparaître une bonne généralité et un ensemble de sous-fonctions pertinent afin de pouvoir évaluer le recul et la maîtrise des candidats en matière de programmation.

Il est à noter également que la part graphique d'un projet n'apportera au mieux qu'un bonus à un projet satisfaisant du point de vue de la modélisation et de l'algorithmique et bien maîtrisé par le candidat.

Le jury est convaincu qu'il est possible d'améliorer certains sujets ou à défaut les solutions apportées (factorisation et création de sous-fonction à bon escient).

Par ailleurs, trop de projets sont encore présentés sans que soient fournies dans les dossiers des traces d'exécution des programmes. Un souci tout particulier doit également être apporté à l'espace de noms utilisé (ensemble des noms de variables et de fonctions). Trop souvent les

projets utilisent des noms de fonctions et de variables non explicites ce qui gêne à la fois la lisibilité des codes sources mais aussi obscurcit les présentations et complique les échanges avec les candidats.

Concernant la qualité des présentations orales des projets elle semble être en progression et les deux stéréotypes pointés dans les rapports précédents moins fréquents. Nous rappelons qu'il faut éviter à la fois une présentation trop générique et un détail du listing des programmes ligne à ligne mais mettre en avant la problématique, la structuration des données, les algorithmes les plus intéressants, la structuration du programme, les tests mis en œuvre pour valider la solution proposée etc. le tout en environ 10 minutes.

Il est particulièrement important que le candidat montre en quoi son implémentation répond au problème posé et qu'il le prouve en explicitant les tests réalisés.

Pour résumer, toute la difficulté de la présentation d'un projet consiste (1) à présenter le problème adressé de manière précise et succincte, (2) à présenter la modélisation informatique du problème, notamment les structures de données permettant de représenter les informations, les opérations réalisées, leur séquençement, (3) à traiter des quelques aspects implantatoires techniques pertinents, notamment les choix réalisés dans l'implantation des algorithmes, etc. et enfin (4) à présenter les exécutions et les tests réalisés ainsi que des points d'amélioration envisagés pour s'assurer dans une certaine mesure du bon fonctionnement du projet et de la maîtrise de celui-ci par le candidat.

Bien entendu, il s'agit des éléments devant se retrouver dans le discours et chaque candidat est libre de structurer son discours en fonction de sa perception du problème et de ses spécificités.

Partie exercice :

Relativement à la partie exercice, nous constatons que le nombre de candidats ne sachant plus faire les premières questions des exercices a diminué tout comme la part de candidats fournissant une solution complète à l'exercice choisi.

Nous constatons, en effet, une tendance au « bachotage ». Les candidats semblant connaître par cœur des "recettes" et essayant de les appliquer dans des cas de figure inappropriés voire proposent de résoudre d'autres exercices que ceux posés. Nous rappelons l'importance d'une bonne lecture des énoncés afin que les candidats puissent profiter au mieux de leur temps de préparation.

Par ailleurs, les éléments suivants restent primordiaux :

- savoir décomposer un problème en un ensemble de problèmes simples pour lesquels les algorithmes standards (recherche, tri, etc.) sont connus.
- savoir choisir une structure de données adaptée à la fois à l'objet d'étude et aux traitements à réaliser.
- avoir une compréhension des algorithmes aussi générique que possible et non dépendante d'une structure de données particulière.

Examineurs : Paolo Ballarini, Guillaume Bouyer, Pierre Connault, Christophe François, Hervé Gras, Céline Hudelot, Antoine Manzanera, Christine Martin (R), Vladimir Paun, Claire Toffano-Nioche.

Expert : Christine Martin

