

PRÉSENTATION GÉNÉRALE

Les oraux d'informatique de la session 2019 du concours Pass' Ingénieur se sont tenus en deux sessions, les 11 et 12 mai 2019 sur le campus de l'ENSIACET à Toulouse et les 25 et 26 mai 2019 sur le campus de Supméca à Saint-Ouen.

Les examinateurs ont noté une grande hétérogénéité des candidats, tant sur leur maîtrise des notions informatiques au programme que sur leur préparation aux modalités de l'épreuve.

Nous reprendrons dans ce rapport quelques rappels sur le déroulement de l'épreuve. Nous effectuerons ensuite quelques remarques sur différents thèmes du programme et leur traitement par les candidats.

Les examinateurs rappellent que quelle que soit la formation universitaire du candidat, celui-ci pourra être interrogé sur l'ensemble du programme. Les candidats sont donc invités à lire attentivement le programme de l'épreuve d'informatique, pour identifier les éventuels points non vus dans leur formation et s'y préparer.

DÉROULÉ DE L'ÉPREUVE

Le candidat se voit remettre un sujet constitué de deux exercices indépendants, portant sur des thèmes différents.

Le candidat dispose de :

- trente minutes pour tenter de résoudre ces exercices, ou du moins en ébaucher les grandes lignes, et préparer sa présentation,
- trente minutes d'oral au tableau pour présenter ses résultats et poursuivre la résolution, avec les questions et indications de l'examineur.

L'épreuve ne comprend pas de partie sur machine : les programmes attendus sont écrits au tableau.

L'oral n'est pas un écrit au tableau : **il est inutile de tout rédiger**. On pourra, par exemple, décrire un programme ou une preuve très similaire à un autre déjà écrit en soulignant les différences à l'oral ; énoncer des hypothèses, ou des prérequis, à l'oral sans les noter... Il est par contre nécessaire de prouver - à l'écrit ou à l'oral - toute affirmation.

Ce bon équilibre entre écrit et oral permet de dynamiser la présentation et permet de tirer le meilleur profit des trente minutes de présentation.

Enfin, les sujets proposés peuvent être volontairement longs. Il n'est pas attendu que le candidat résolve toutes les questions, mais qu'il aborde durant sa préparation au moins une partie de chaque exercice et soit capable de rebondir sur les indications de l'examineur.

REMARQUES SUR LES QUESTIONS DE PROGRAMMATION

Sur les deux exercices du sujet l'un au moins comprend des questions de programmation, pour lesquelles un code est attendu dans un langage de programmation donné.

Le candidat est invité, en début d'épreuve, à choisir librement son langage – et à s'y tenir. Sur ce choix, les examinateurs conseillent de choisir le langage sur lequel le candidat est le plus à l'aise, sans calcul : les examinateurs ont le souci de proposer des questions s'adaptant à tout langage et le choix d'un langage donné n'est pas pénalisant.

Cette année, les candidats ont opté pour le C, le java et le python ; sans que les examinateurs ne remarquent une meilleure maîtrise des candidats ayant opté pour tel ou tel langage.

Les examinateurs notent une grande hétérogénéité de maîtrise de la syntaxe des langages. Il n'est pas nécessairement attendu un code parfaitement correct, qui compile, mais le candidat est toutefois attendu sur sa maîtrise d'un langage donné, sans mélange ni pseudo-code.

Enfin, la clarté et la simplicité du code sont aussi des éléments pris en compte. Dans l'ensemble, les candidats de cette session ont plutôt bien indenté et ordonné leurs programmes.

REMARQUES SUR LES QUESTIONS D'ALGORITHME

Les techniques algorithmiques classiques telles que les algorithmes gloutons, diviser pour régner, la programmation dynamique ou le backtracking sont supposées connues des candidats. Les sujets pourront amener le candidat à appliquer une méthode connue pour un problème nouveau pour lui.

En outre, il est attendu des candidats qu'ils n'aient pas de difficulté à implémenter des algorithmes de base tels que des algorithmes de somme, de maximum, de comptage, de recherche, de tri - adaptés à un contexte donné.

Les examinateurs notent une aptitude souvent décevante des candidats à résoudre un problème nouveau par ces techniques classiques.

Les candidats ont aussi très souvent du mal à illustrer le déroulement d'un algorithme sur une entrée donnée. Cela est parfois l'objet d'une question : dérouler l'exécution d'un programme sur un exemple donné ; mais c'est aussi souvent un bon choix de s'appuyer sur un ou des exemples pour concevoir ou expliquer un algorithme.

Pour ce qui est de l'analyse d'algorithmes, l'analyse de la complexité d'un programme est souvent approximative. Si l'analyse d'un programme simple avec plusieurs boucles imbriquées est souvent réussie, les candidats négligent souvent la complexité des appels de fonctions et sont souvent bloqués sur des calculs plus complexes ;

enfin pour des problèmes connus tels que le tri, les examinateurs notent de régulières confusions entre la complexité du problème (en $O(n \log(n))$) et la complexité de l'algorithme proposé (par exemple en $O(n^2)$).

REMARQUES SUR LES DIFFÉRENTS POINTS D'INFORMATIQUE THÉORIQUE

Les domaines des graphes, des automates, de la logique propositionnelle, des arbres ont aussi donné lieu à des questions plus théoriques, pour lesquelles il n'était pas demandé de code. Ces domaines en particulier ne doivent pas faire l'objet d'impasse de la part des candidats.

Les candidats ne seront pas attendus principalement sur le résultat, mais sur la méthode mise en œuvre et sa présentation.

En logique booléenne, certaines questions sont résolubles de multiples façons : par exemple, déterminer que deux formules sont équivalentes. Le candidat est alors libre de choisir la méthode qui lui semble la plus adaptée et la plus rapide.

Au-delà des questions plus techniques et attendues telles que le calcul d'une table de vérité d'une formule booléenne, ou la détermination d'un automate, il est attendu des candidats une compréhension des objets et de leur signification. Ainsi par exemple, certains candidats sachant déterminer un automate se sont trouvés bloqués pour décrire le langage reconnu, même simple ; ou pour proposer un automate pour un langage donné.