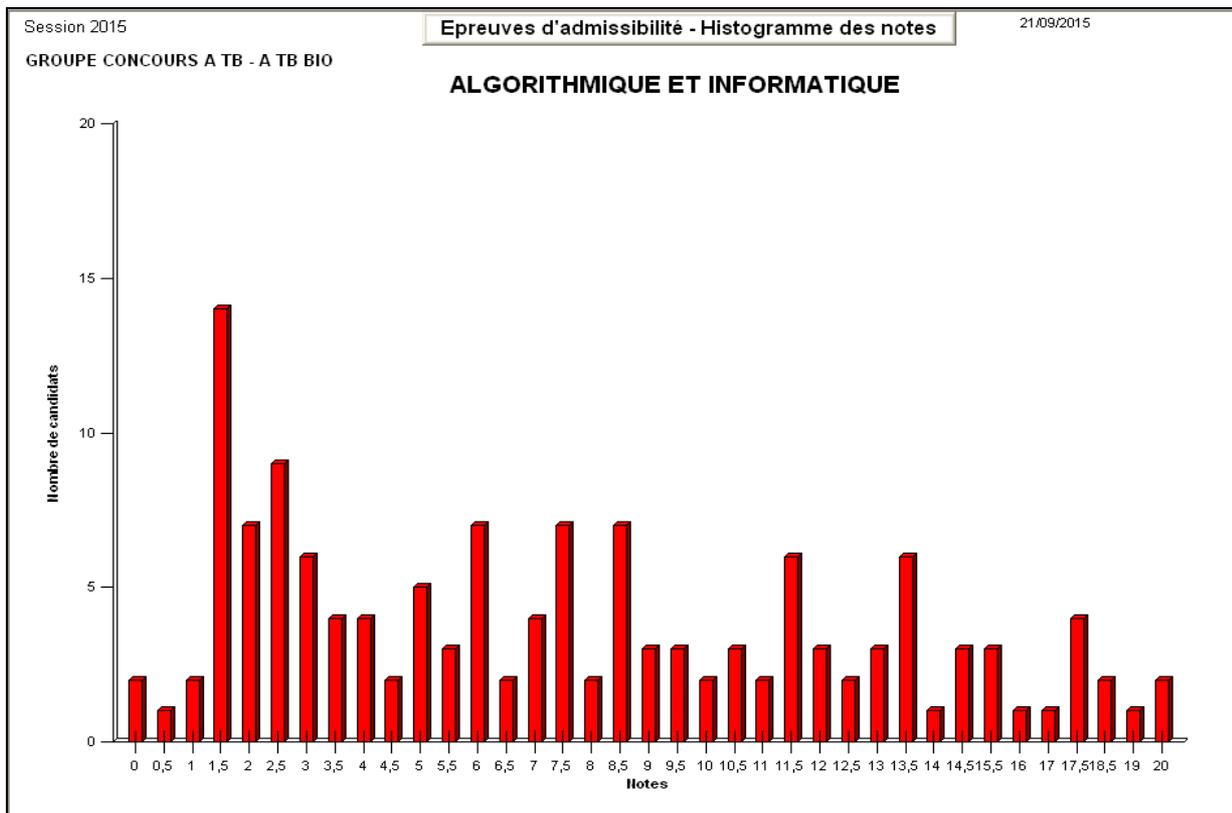


Épreuve écrite d'Algorithmique et Informatique

Concours	Nb.cand	Moyenne	Ecart type	Note la plus basse	Note la plus haute
TB BIO	139	7,5	5,1	0,0	20
TB ENV	78	7,03	5,2	0,0	20
POLYTECH TB	129	7,7	5,1	0,5	20

Il y avait 143 candidats qui ont obtenu des notes entre 0 et 20, avec une moyenne de 7,51 et un écart-type de 5,1.



Description de l'épreuve

L'épreuve, d'une durée de 45 minutes, se compose de deux parties indépendantes autour de la simulation de lancers de pièces. Les différentes parties du programme ont été abordées.

Analyse de l'épreuve

La première partie comporte des questions classiques d'écriture de fonctions (en langage *Python* exclusivement) en évaluant principalement leurs compétences en algorithmique : simulations (de variables aléatoires), structures de boucles, manipulation du type *list*, calcul de moyenne, etc. La difficulté est croissante sur cette partie, en commençant par une question largement abordable.

1. La plupart des candidats réussit cette question facile (compléter un code partiellement écrit) mais sans réussir à obtenir la totalité des points. Le jury rappelle que la simulation d'une expérience de Bernoulli est un attendu du programme.

2. La question est plutôt réussie.
3. Un candidat sur deux n'a pas réussi la question : beaucoup maîtrisent encore mal l'utilisation d'une boucle *while*.
4. Il s'agissait de répéter *n* fois l'expérience simulée dans la question précédente et stocker les résultats dans une liste, ce qui a posé problème à plus de la moitié des candidats.
5. La question est classique et plutôt bien traitée.
6. La question n'a pas souvent été traitée alors qu'il s'agit d'un attendu explicite du programme. Les rares étudiants ayant traité la question l'ont généralement bien réussie.

La deuxième partie du sujet comportait un questionnaire à choix multiples (QCM) ainsi qu'une dernière question demandant de modifier un algorithme fourni précédemment. Le QCM permettait de tester la compréhension par le candidat de codes qu'il n'a pas conçus, tandis que la dernière question, voulue volontairement plus difficile, permettait de distinguer les meilleurs candidats.

1. (a. et b.) Ces deux questions sont les plus faciles du sujet (inférence de type et prévision de résultats). Sans surprise, elles ont été les mieux réussies.
2. Il s'agit d'analyser le code de quatre fonctions et de déterminer celles qui vérifient une propriété donnée. La question est réussie par un tiers des candidats ; cela peut s'expliquer par le nombre de structures et syntaxes différentes à analyser dans les différentes fonctions.
3. Il s'agit ici de prévoir la valeur renvoyée par une instruction (appel de fonction). La question rapport en moyenne la moitié des points aux candidats.
4. Cette dernière question (difficile) a été la moins traitée (16% des candidats). Parmi eux, beaucoup n'ont pas compris la question. Les candidats les plus brillants se sont vus récompensés de plus de deux tiers des points.

Le jury a relevé plusieurs points lors de la correction des copies :

- le sujet a semblé adapté à la durée de l'épreuve ; les candidats sont donc invités à se relire pour éviter des erreurs d'étourderies évidentes ;
- la première partie a souvent été comprise de manière approximative ; ainsi, beaucoup de candidats n'ont traité que peu de questions, alors qu'il semblait aisé d'obtenir la moyenne sur des questions classiques et dans l'esprit du programme ;
- de nombreuses erreurs algorithmiques et liées au langage *Python* ont été relevées :
 - les erreurs de typage et de logique sur les structures de contrôle (boucles et tests) sont lourdement sanctionnées. Il est rappelé aux candidats que la rigueur nécessaire à cette épreuve est au moins aussi importante que celle appliquée aux autres épreuves scientifiques ;
 - la confusion entre la fonction *print* et le mot-clé *return* est très fréquente. Il est nécessaire de comprendre la distinction entre les deux instructions, notamment que la fonction *print* n'est utilisée qu'à la seule condition d'afficher une valeur, sans que cette dernière ne puisse être par la suite un argument d'un calcul.

Correcteurs : *FRANÇOIS Christophe, PRÉAUX Jean-Philippe*
Concepteur du sujet : *HAÏDAR Kamal*