

Commentaires sur le contenu de l'épreuve commune d'Informatique du concours d'admission à l'École polytechnique

L'épreuve consiste à vérifier la compréhension des notions fondamentales et élémentaires de l'algorithmique et de la programmation acquises dans le cadre du programme d'informatique commun aux classes préparatoires. On testera la précision de la pensée algorithmique et sa mise en oeuvre dans un noyau de langage isomorphe au noyau de tout langage de programmation moderne.

Le langage de programmation est laissé au choix parmi Maple, Mathematica ou l'un des langages suivants: Ada, Caml, C, C++, Java, Modula, Pascal, Ocaml. Si les détails de la syntaxe ne sont pas considérés comme importants, le langage support choisi sera précisé et on n'utilisera que les opérations de base, et non les fonctions puissantes de sa bibliothèque. Les seules fonctions graphiques autorisées permettront le tracé des segments entre deux points d'un repère cartésien du plan: `MoveTo(x, y)`, `LineTo(x, y)`, `Line(dx, dy)`.

L'épreuve portera plus particulièrement sur les points suivants (présents dans la partie Algorithmes et Programmation des programmes actuels):

- *structures des programmes:*
déclarations et instructions;
instructions de contrôle (séquence, instruction conditionnelle, instructions d'itération);
fonctions et procédures;
variables locales, variables globales;
types de base: entiers, booléens, flottants;
tableaux multidimensionnels;
- *méthodes algorithmiques itératives:*
parcours séquentiel d'un tableau (calculs du minimum, du maximum d'un tableau, produit scalaire de deux vecteurs, évaluation d'un polynôme, etc);
double itération dans un tableau (tris élémentaires, etc);
recherche linéaire en table;
itération numérique (limites de suites convergentes, calcul de dérivées, recherche de zéros d'une fonction continue, etc);
calcul matriciel (addition, multiplication, lecture/impression, méthodes du pivot);
- *méthodes algorithmiques récursives:*
calcul de fonctions récursives (factorielle, fibonacci),
méthodes dichotomiques (encadrement de racines, recherche en table);
exemples de courbes fractales élémentaires;

On se limitera à des raisonnements simples pour l'évaluation de la complexité des algorithmes: il s'agira d'être capable de compter le nombre d'opérations dans des boucles itératives. Il n'est pas question de faire des évaluations élaborées faisant intervenir la théorie de la complexité.

L'épreuve est principalement axée sur l'écriture de petits programmes (d'environ moins de 20 lignes) et l'apprentissage de la pensée algorithmique. On pourra faire référence aux algorithmes numériques utilisés dans les travaux pratiques ou les travaux dirigés de chimie, mathématiques, physique ou sciences industrielles, effectués dans les classes préparatoires. Le programme de l'épreuve ne contient aucune notion d'informatique théorique, comme la théorie des automates finis ou du calcul logique propositionnel. Les structures d'enregistrement (*records*) et les structures de données dynamiques (listes, arbres) ne sont pas au programme. De même, les algorithmes sur les graphes ne seront pas considérés. On veillera particulièrement à ne pas coder les structures de données dynamiques dans des tableaux, en

remplaçant les références (ou pointeurs) par des indices dans ces tableaux.

Le programme de l'épreuve ne fait pas appel à des notions complexes d'algorithmique. Par exemple, la partie récursivité devra être bien différenciée de la méthode algorithmique ``Diviser pour Régner''. Il s'agit plus de montrer que les calculs par récurrence peuvent s'écrire avec des fonctions numériques récursives, ou que l'écriture récursive doit être utilisée quand elle est plus naturelle (par exemple dans la recherche dichotomique, ou dans la multiplication avec l'algorithme de Horner). Dans cette partie, aucune connaissance ne sera a priori exigée, mais les candidats devront être familiers avec l'existence de l'écriture récursive.

Comme point de repère, le programme de cette nouvelle épreuve correspond au cours de la semaine d'initiation à l'informatique, introduite à l'Ecole Polytechnique depuis 1999 (cf. <http://www.enseignement.polytechnique.fr/informatique/Old/Init0/semaine0.html>).

Les ouvrages suivants peuvent servir de base à la préparation d'un cours mais contiennent pour la plupart infiniment plus que ce qui est demandé dans le cadre de cette épreuve.

- **[d]** Algorithms & Data Structures, Niklaus Wirth, Prentice-Hall, 1986.
- **[d]** Mathématiques et Informatique, Jean Berstel, Jean-Eric Pin, Michel Pocchiola, McGraw-Hill, 1991.
- **[d]** Algorithms, Robert Sedgewick, Addison-Wesley, 1988. En français: Algorithmes en langage C, trad. par Jean-Michel Moreau, InterEditions, 1991.
- **[g]** The New Turing Omnibus, 66 Excursions in Computer Science, A. K. Dewdney, Computer Science Press, 1993.
- **[e]** Programming Pearls, Jon Bentley, Addison-Wesley, 1989.
- **[g]** Algorithmics, The Spirit of Computing, David Harel, Addison-Wesley, 1993.
- **[r]** Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, MIT Press, 1990.

[g]= culture générale; **[d]**= didactique; **[r]**= ouvrage de référence; **[e]**= exercices.

This document was translated from $L^A T_E X$ by [H^EV^EA](#).