

Algorithmique avancée et programmation C
Référentiel de compétences disciplinaires

Domaines	Sous domaines	Code	Compétences disciplinaires	À maîtriser pour le partiel
Analyse	Analyse descendante	AN001	Désigner les choses (identifiant significatif)	X
		AN002	Être précis quant aux types de données utilisés	X
		AN003	Connaître le rôle de l'analyse	X
		AN004	Comprendre et savoir appliquer des consignes algorithmiques sur un exemple	X
		AN101	Identifier les entrées et sorties d'un problème	X
		AN102	Décomposer logiquement un problème	X
		AN103	Généraliser un problème	X
		AN104	Savoir si un problème doit être décomposé	X
		AN105	Identifier un problème naturellement récursif (directement ou indirectement)	X
		Type abstrait de données	AN201	Identifier les dépendances d'un TAD
	AN202		Définir des TAD génériques	X
	AN203		Savoir si une opération identifiée fait partie du TAD à spécifier	X
	AN204		Formaliser des opérations d'un TAD	X
	AN205		Formaliser les préconditions d'une opération d'un TAD	X
	AN206		Formaliser des axiomes ou savoir définir la sémantique d'une opération d'un TAD	X
	Collection	AN301	Lister les collections usuelles	X
		AN302	Formaliser sous forme de TAD une collection	X
	Graphe	AN401	Spécifier les différents types de graphes (étiqueté et/ou valué)	
	Conception préliminaire	CP001	Comprendre le paradigme de programmation impératif	X
		CP002	Comprendre le paradigme de programmation structuré	X
CP003		Choisir entre une fonction et une procédure	X	
CP004		Concevoir une signature (préconditions incluses)	X	
CP005		Choisir un passage de paramètre (E, S, E/S)	X	
CP006		Comprendre le rôle de la conception préliminaire	X	
Conception détaillée	CD001	Dissocier les deux rôles du développeur : concepteur et utilisateur	X	
	CD002	En tant qu'utilisateur, respecter une signature	X	
	CD003	Utiliser le principe d'encapsulation	X	

Algorithmique avancée et programmation C
Référentiel de compétences disciplinaires

Domaines	Sous domaines	Code	Compétences disciplinaires	À maîtriser pour le partiel
		CD004	Écrire des algos avec le pseudo code utilisé à l'IN-SA	X
		CD005	Écrire un pseudo code lisible (court, indentaté, avec des identifiants significatifs)	X
		CD006	choisir la bonne itération	X
		CD007	Utiliser les bonnes catégories de paramètres effectifs pour un passage de paramètre donnée	X
		CD009	Écrire un algorithme qui résout le problème	X
		CD010	Connaître le rôle de la conception détaillée	X
		CD011	utiliser les bons types de données (paramètres formels, variables locales)	X
	Complexité	CD101	Estimer la taille d'un problème (n)	X
		CD102	Calculer une complexité dans le pire et le meilleur des cas	X
		CD103	Exprimer une complexité en temps et en espace	X
		CD104	Écrire un algorithme d'une complexité donnée	X
	Récurtivité	CD201	Identifier et résoudre le problème des cas non récurtifs	X
		CD202	Identifier et résoudre le problème des cas récurtifs	X
		CD203	Identifier une récurtivité terminale et non terminale et ce que cela implique	X
	Dichotomie	CD301	Identifier un problème qui se résout à l'aide d'un algorithme dichotomique	X
		CD302	Définir l'espace de recherche d'un algorithme dichotomique	X
		CD303	Savoir diviser et extraire les bornes de l'espace de recherche d'un algorithme dichotomique (cas discret ou continu)	X
	SDD	CD401	Concevoir et utiliser des listes chaînées	X
		CD402	Concevoir et utiliser des listes doublement chaînées	X
		CD403	Concevoir et utiliser des arbres (binaires, n-aires)	X
		CD404	Concevoir et utiliser des arbre-B (insertion suppression)	
	Tris	CD501	Comprendre les algorithmes des différents tris et leurs complexités	X
	Collection	CD601	Concevoir des collections à l'aide de SDD	
		CD602	Comprendre les algorithmes d'insertion et de suppression (naïfs et AVL) dans un arbre binaire de recherche	

Algorithmique avancée et programmation C
Référentiel de compétences disciplinaires

Domaines	Sous domaines	Code	Compétences disciplinaires	À maîtriser pour le partiel
	Programmation dynamique	CD701 CD702 CD703	Définir la programmation dynamique Appliquer la programmation dynamique pour des cas simples Comprendre et résoudre le problème du sac à dos	
	Graphe	CD801 CD802 CD803 CD804	Concevoir des graphes (matrice d'adjacence, matrice d'incidence, liste d'adjacence) Écrire des algorithmes de parcours en largeur ou en profondeur Écrire un algorithme de tri topologique Comprendre des algorithmes de recherche du plus court chemin : Dijkstra et A*	
	Représentation d'un TAD	CD901	Concevoir un type de données adapté à la situation en terme d'espace mémoire et d'efficacité	X
Développement C		DEV001 DEV002 DEV003 DEV004 DEV005 DEV006 DEV007 DEV008 DEV009 DEV010 DEV011 DEV012	Compiler et linker un programme C (options de base de gcc) Débugger un programme Développer des modules C (.h et .c) Écrire un Makefile simple Créer des bibliothèques statiques ou dynamiques Écrire un code C lisible Traduire/adapter un algorithme en fonction C Traduire des passages de paramètre algorithme en passage de paramètre C Utiliser les pointeurs, tableaux et chaînes de caractères Appliquer une méthodologie de développement Développer des SDD génériques (void*) Utiliser et passer en paramètre des fonctions	X X X X X X X X X X X
Tests unitaires		TU001	Écrire des tests unitaires à l'aide du framework Cunit	X