

Algorithmique avancée et programmation C
Référentiel de compétences

Domaine de compétences	Sous domaine de Compétences	Code	Compétence	À maîtriser pour le partiel
Analyse	Analyse descendante	AN001	Savoir désigner les choses (identifiant significatif)	X
		AN002	Savoir être précis quant aux types de données utilisés	X
		AN003	Connaître le rôle de l'analyse	X
		AN101	Savoir identifier les entrées et sorties d'un problème	X
		AN102	Savoir décomposer logiquement un problème	X
		AN103	Savoir généraliser un problème	X
		AN104	Savoir si un problème doit être décomposé	X
	Type abstrait de données	AN105	Savoir identifier un problème naturellement récursif (directement ou indirectement)	X
		AN201	Savoir identifier les dépendances d'un TAD	X
		AN202	Savoir définir des TAD générique	X
		AN203	Savoir si une opération identifiée fait partie du TAD à spécifier	X
		AN204	Savoir formaliser des opérations d'un TAD	X
		AN205	Savoir formaliser les préconditions d'une opération d'un TAD	X
	Collection	AN206	Savoir formaliser des axiomes ou savoir définir la sémantique d'une opération d'un TAD	X
		AN301	Savoir lister les collections usuelles	X
		AN302	Savoir formaliser sous forme de TAD une collection	X
	Graphe	AN401	Savoir spécifier les différents types de graphes (étiqueté et/ou valué)	
Conception préliminaire		CP001	Savoir ce que représente le paradigme de programmation impératif	X
		CP002	Savoir ce que représente le paradigme de programmation structuré	X
		CP003	Savoir choisir entre une fonction et une procédure	X
		CP004	Savoir concevoir une signature (préconditions incluses)	X
		CP005	Savoir choisir un passage de paramètre (E, S, E/S)	X
		CP006	Connaître le rôle de la conception préliminaire	X
Conception détaillée		CD001	Savoir dissocier les deux rôles du développeur : concepteur et utilisateur	X
		CD002	En tant qu'utilisateur, savoir respecter une signature	X
		CD003	Savoir utiliser le principe d'encapsulation	X
		CD004	Savoir écrire des algos avec le pseudo code utilisé à l'INSA	X
		CD005	Savoir écrire un pseudo code lisible (indentation, identifiant significatif)	X

Algorithmique avancée et programmation C
Référentiel de compétences

Complexité	CD006	Savoir choisir la bonne itération	X	
	CD007	Savoir quelles catégories de paramètres effectifs peut être utilisé avec un passage de paramètre données	X	
	CD009	Savoir écrire un algorithme qui résout le problème	X	
	CD010	Connaître le rôle de la conception détaillée	X	
	CD101	Savoir estimer la taille d'un problème (n)	X	
	CD102	Savoir calculer une complexité dans le pire et le meilleur des cas	X	
	CD103	Savoir exprimer une complexité en temps et en espace	X	
	CD104	Savoir écrire un algorithme d'une complexité donnée	X	
	Récursivité	CD201	Savoir identifier et résoudre le problème des cas non récursifs	X
		CD202	Savoir identifier et résoudre le problème des cas récursifs	X
CD203		Savoir identifier une récursivité terminale et non terminale et ce que cela implique	X	
CD204		Savoir écrire un algorithme qui respecte une complexité donnée	X	
Dichotomie	CD301	Savoir identifier un problème qui se résout à l'aide d'un algorithme dichotomique	X	
	CD302	Savoir 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 continue)	X	
SDD	CD401	Savoir concevoir et utiliser des listes chaînées	X	
	CD402	Savoir concevoir et utiliser des listes doublement chaînées	X	
	CD403	Savoir concevoir et utiliser des arbres (binaires, n-aires)	X	
	CD404	Savoir concevoir et utiliser des abres-B (insertion suppression)		
Tris	CD501	Connaître les algorithmes des différents tris et leurs complexités	X	
Collection	CD601	Savoir concevoir des collections à l'aide de SDD		
	CD602	Connaître les algorithmes d'insertion et de suppression (naïfs et AVL) dans un arbre binaire de recherche		
Programmation dynamique	CD701	Savoir définir la programmation dynamique		
	CD702	Savoir appliquer la programmation dynamique pour des cas simples		
	CD703	Connaître et savoir résoudre le problème du sac à dos		

Algorithmique avancée et programmation C
Référentiel de compétences

	<p>Graphe</p> <p>Représentation d'un TAD</p>	<p>CD801</p> <p>CD802</p> <p>CD803</p> <p>CD804</p> <p>CD901</p>	<p>Savoir concevoir des graphes (matrice d'adjacence, matrice d'incidence, liste d'adjacence)</p> <p>Savoir écrire des algorithmes de parcours en largeur ou en profondeur</p> <p>Savoir écrire un algorithme de tri topologique</p> <p>Connaître des algorithmes de recherche du plus court chemin : Dijkstra et A*</p> <p>Savoir concevoir un type de données adapté à la situation en terme d'espace mémoire et d'efficacité</p>	<p>X</p>
Développement C		<p>DEV001</p> <p>DEV002</p> <p>DEV003</p> <p>DEV004</p> <p>DEV005</p> <p>DEV006</p> <p>DEV007</p> <p>DEV008</p> <p>DEV009</p> <p>DEV010</p> <p>DEV011</p> <p>DEV012</p>	<p>Savoir compiler et linker un programme C (options de base de gcc)</p> <p>Savoir débogger un programme</p> <p>Savoir développer des modules C (.h et .c)</p> <p>Savoir écrire un Makefile simple</p> <p>Savoir créer des bibliothèques statiques ou dynamiques</p> <p>Savoir écrire un code C lisible</p> <p>Savoir traduire/adapter un algorithme en fonction C</p> <p>Savoir traduire des passages de paramètre algorithme en passage de paramètre C</p> <p>Maîtriser les pointeurs, tableaux et chaînes de caractères</p> <p>Savoir appliquer une méthodologie de développement</p> <p>Savoir développer des SDD génériques (void*)</p> <p>Savoir utiliser et passer en paramètre des fonctions</p>	<p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p>
Tests unitaires		<p>TU001</p>	<p>Savoir écrire des tests unitaires à l'aide du framework Cunit</p>	<p>X</p>