

TABLE DES MATIÈRES

I Informations pratiques	11
1 Généralités	13
1.1 Objectifs de l'enseignement	13
1.2 Où trouver	14
1.2.1 les polycopiés	14
1.2.2 le site internet	14
1.2.3 les profs	14
2 Le principe de la notation	17
2.1 La présence en cours et en TPs	17
2.2 Le contrôle continu	17
II Le cours	19
3 Introduction	23
3.1 Notions d'algorithmes	23
3.2 Programmation	24
3.2.1 Programmation linéaire :	24
3.2.2 Récursivité :	26
3.2.3 Programmation non-linéaire	27

4	Les bases du langage C	29
4.1	Nature & composantes	29
4.1.1	Introduction	29
4.1.2	Hierarchie du langage C :	31
4.1.3	Anatomie d'un programme C	32
4.2	Les variables en C	33
4.2.1	Types et définitions	33
4.2.2	Déclaration de variables : syntaxe	33
4.2.3	Déclarations de variables et leur portée	34
4.2.4	Entier ou caractère .. ?	35
4.2.5	Les constantes : propriétés, notation	35
4.2.6	Conversion de types, le cast	37
4.3	Les opérateurs du langage	38
4.3.1	Les différents opérateurs	38
4.3.2	Pseudo-code du TD 1	39
4.3.3	Opérateurs du langage : TD 1	40
4.3.4	Priorité des opérateurs	40
4.3.5	Exemple de programme C	41
4.4	Instructions de branchement	42
4.4.1	Branchement conditionnel	42
4.4.2	Branchement non-conditionnel	44
4.5	Les affectations composées	45
4.6	Les fonctions en C	46
4.6.1	Définition intuitive :	46
4.6.2	Déclaration de fonctions : le patron	46
4.7	Structure d'un programme	48
4.7.1	Liste de directives au compilateur C	49
5	Le système d'exploitation Unix/Linux	51
5.1	Caractéristiques principales d'Unix	51
5.2	Outils de travail : logiciels	52
5.3	Système de fichiers	52
5.4	Fichiers et répertoires	53
5.5	Les utilisateurs	53
5.6	Les droits d'accès	54
5.7	Visualiser l'info sur les fichiers	54
5.8	Changer les droits d'accès	55
5.9	Les commandes les plus courantes	55
5.10	Enchaînement des commandes	57
5.11	Les scripts shell	58
5.11.1	Set, echo, cat	59

	5
5.11.2	Branchement conditionnel : foreach, while 60
5.11.3	Exemple d'un C-shell 61
6	Mieux articuler le langage C 63
6.1	Les types composés & personnalisés 63
6.2	Formats d'entrées/sorties en C 67
6.2.1	Saisir des données formatées sur entrée standard 68
6.2.2	Écrire des données formatées sur sortie standard 70
6.2.3	Lire et écrire des données non-formatées sur e/s standards 71
6.3	Les pointeurs : définition, propriétés 72
6.3.1	Le pointeur 73
6.3.2	Initialisation d'un pointeur. 73
6.3.3	Opérateur & 74
6.3.4	L'allocation d'espace mémoire 74
6.3.5	Libérer l'espace mémoire 74
6.3.6	La valeur pointée 75
6.3.7	Notation arithmétique des pointeurs 75
6.3.8	Conversion des pointeurs 75
6.3.9	Pointeurs et fonctions 76
6.3.10	Applications de pointeurs 77
6.4	Créer une bibliothèque 82
6.4.1	Second regard sur la compilation 82
6.4.2	Compiler les objets de fonctions 82
6.4.3	Archiver ses fonctions 83
6.4.4	Édition de liens 83
7	Entrées et sorties en C 87
7.1	Lire et écrire des données dans des fichiers personnels 87
7.2	Gestion des entrées/sorties : notion de tampon 88
7.3	Ouverture et fermeture de fichiers 89
7.4	Quelques opérations de lecture/écriture 90
7.5	Lecture/écriture formatées 91
7.5.1	Se positionner dans un fichier 91
7.5.2	Saisir des paramètres à la ligne de commandes 92
III	Travaux Dirigés 95
1	Algorithmes et pseudo-codes 97
2	Expressions et entiers 101

6	
3	Macro-molécules et zéros de fonctions 103
4	Tri et évaluation de potentiel 107
5	Itérations 111
IV	Examens de l'année précédente 113
V	Outils pour la pratique 121
1	Les bases des séances de TP 123
1.1	Comment ouvrir une session ? 123
1.2	Premiers pas 124
1.3	Comment ouvrir un éditeur de texte ? 125
1.3.1	Comment sauvegarder un fichier ? 126
1.4	Comment ouvrir un navigateur web ? 126
1.5	Comment compiler un programme ? 127
2	Aides techniques 129
2.1	Rappel des commandes Unix les plus courantes 129
2.1.1	commandes utilitaires 129
2.1.2	commandes sur les fichiers 129
2.1.3	Filtres évolués 130
2.1.4	Droits d'accès aux fichiers 130
2.1.5	Processus 131
2.1.6	Compression, encodage, archivage 131
3	"Et quand ça ne marche pas ?" 133
3.1	L'erreur la plus fréquente : parse error 133
3.2	Fautes de frappes 133
3.2.1	Mélange entre = et == 133
3.2.2	Priorité des opérateurs 134
3.3	Oubli de la condition de sortie 134
3.4	Variables locales et globales 134
3.5	Allocation de la mémoire : segmentation fault 135
3.6	Entrées/sorties standard 135
3.6.1	scanf 135
3.6.2	printf 135
3.7	Debuggons ensemble 135
3.8	Avertissement sur les messages d'erreur 137

3.9	Et si ça ne marche toujours pas ?	137
3.10	Et si ca marche vraiment toujours pas et que j'ai tout essayé ? . . .	137
4	Petite bibliographie	139
VI	Travaux pratiques	141
1	Unix et premiers programmes	143
2	Factorielle, Algorithme de Hoerner	147
3	Fonctions et bibliothèques	151
4	Entrées et sorties	155