

Appendices

Références

.1 Bibliographie

- [RJLK] RITCHIE, Dennis M. and JOHNSON, S.C. and LESK, M.E. and KERNIGAN, B.W., *The C Programming Language*, **Bell System Technical Journal**, vol. 57, 1978, pp. 1991-2019.
- [Présente les origines historiques de C ainsi que ses avantages et ses faiblesses. Le langage C a été défini par Dennis RITCHIE (des Bell Laboratories) en 1972. C'est un descendant du langage ALGOL en passant par les langages CPL, BCPL et B (d'où son nom).]
- [KR-79] KERNIGHAN, Brian and RITCHIE, Dennis, **The C Programming Language**, Prentice-Hall, 1979, X+228 p. Tr. fr. par Thierry BUFFENOIR, **Le langage C**, préface de G. ROUCAIROL, Masson, 1983, IX + 218 p.
- [La référence sur le langage C, en fait la version K&R (d'après les initiales des auteurs), bien que pas facile à lire.]
- [KR-88] KERNIGHAN, Brian and RITCHIE, Dennis, **The C Programming Language (second edition)**, Prentice-Hall, 1988. Tr. fr. par Jean-François GROFF et Éric MOTTIER, **Le langage C : C ANSI**, Masson, 1990, XII+280 p.
- [Correspond au C ANSI.]
- [ANS-89] **American National Standard X3.159**, 1989.
- [La définition du standard ANSI C.]
- [Pla-92] P. J. PLAUGER, **The Standard C library**, Prentice Hall, 1992. Tr. fr. par Isabelle SAINT-SAËNS, **La bibliothèque C standard**, Masson, 1995, XIV + 518 p.
- [TG-] Clovis L. TONDO and Scott E. GIMPEL, **The C. Answer book**, Prentice Hall, 1989. tr. fr. par Antoine BERTIER, **Le langage C : C ANSI, solutions**, Masson, 1990, X + 157 p.
- [Solutions aux exercices de [KR-88].]

Index

- * (symbole de multiplication), 34
- `.lib`, 97
- `#define`, 58
- % (comme opérateur), 40
- `%E`, 38
- `%G`, 38
- `%Ld`, 51
- `%c`, 51
- `%d`, 34
- `%e`, 38
- `%f`, 37
- `%g`, 38
- `%s`, 57
- `\a`, 43
- `\b`, 43
- `\f`, 43
- `\n`, 43
- `\r`, 43
- `\t`, 43

- `a.out` (*Assembler Output*), 28
- abaciste, 6
- `abs()`, 39
- `acos()`, 41
- addition, 3
- affectation, 48
 - combinée, 73
 - en chaîne, 73
- affecter, 48
- AL-KHWARIZMI, Mohammed ibn-Musa, 6
- algoriste, 6
- algorithme, 6, 78
- alternative, 70
- ampersand*, 56
- backslash*, 51
- appel
 - de fonction, 86
- approchée, valeur, 9
- `asin()`, 41
- `atan()`, 41
- avertissement, 32

- BABBAGE, Charles (1791–1871), 15
- base, 3
- binder*, 32
- blanc, 43
- bloc, 30, 62
- booléen, 63
- boucle, 66
- bureau de calcul, 16

- C++, langage, 25
- C, langage, 25
- cahier des charges, 78
- calcul, 3
 - définition précise, 80
- calculette, 29
 - alphanumérique, 29
 - programmable, 29
 - quatre opérations, 29
 - scientifique, 29
- call*
 - by reference*, 91
 - by value*, 91
- caractère
 - non affichable, 101
 - spécial, 49
- cast*, 54
- `cc` (*C Compiler*), 28
- `ceil()`, 41
- chaîne de caractères, 57
- `char`, 50
- chiffre, 4
 - arabe, 6
- `clrscr()`, 104
- codage, 78
- commande, 32
- commentaire, 77
- compilateur, 25
 - GNU, 27
- compilation, 24
- compter, 2
- conception

- en deux phases, 78
- `conio.h`, 102
- `const`, 59
- constante, 58
- constructeur
 - de types, 52
- contre-oblique, 51
- conversion
 - de type, 54
- corps
 - d'une boucle, 66
 - de fonction, 85
- `cos()`, 41
- `cosh()`, 41
- curseur, 55
- déclaration
 - d'une fonction
 - globalement, 95
 - localement, 95
 - de fonction, 94
 - de variable, 48
- décrémentation, 73
- dénombrer, 2
- dépassement de capacité, 44
- directive
 - de format, 56
- diséquation, 64
- DJGPP, 28
- double, 51
- duplication, 4
- éditeur
 - de lien, 32
 - de texte, v
- `emacs`, 27
- en-tête
 - d'une boucle, 66
 - de fonction, 85
- erreur
 - de sémantique, 79
 - de syntaxe, 30, 79
- espace, 43
- esperluette, 56
- exécution
 - d'un programme, 24
- `exp()`, 41
- exposant, 38
- expression
 - booléenne, 63
 - numérique, 42
- `fabs()`, 41
- fatal error*, 32
- FIBONACCI, 6
- fichier
 - binaire, 25
 - en-tête, 31
 - exécutable, 25, 32
 - inclus, 31
 - objet, 32
 - texte, 25
- `float`, 51
- `floor()`, 41
- fonction, 83
 - prédéfinie, 31
 - prédéfinie, 83
- format, 34
- forme
 - infixe, 64
 - suffixe, 64
- fraction, 10
 - unitaire, 10
- `gcc` (*Gnu cc*), 28
- Gerbert d'Aurillac, 6
- `getch()`, 102
- `getchar()`, 102
- `getche()`, 102
- guillemet
 - vertical, 30
- header file*, 31
- identificateur, 49
- implémentation, 78
- `include` (commande), 31
- incrémentation, 73
- indentation, 76
- indexation
 - indirecte, 98
- initialisation
 - d'une boucle, 67
 - des variables, 52
- instruction, 30
 - bloquante, 56
 - composée, 61
 - élémentaire, 61
 - simple, 61
 - vide, 62

- int, 50
- interprétation, 24
- interpréteur de commande, v
- invite, 56
- itération, 66
- L, 51
- langage
 - C, 25
 - C++, 25
 - d'assemblage, 24
 - de programmation, 24
 - évolué, 24
 - machine, 24
- langue
 - naturelle, 24
- linker*, 32
- Linux, 27
- log(), 41
- log10(), 41
- logiciel, 24
- logistique, 13
- long, 50
- LOVELACE, Lady Ada, 15
- Lvalue*, 73
- main(), 30
- mantisse, 38
- math.h, 39
- médiation, 4
- méthode
 - de Hörner, 103
- méthodologie, 76
- mise au point
 - d'un programme, 55
- modificateur, 50
- modularité, 83
- mot
 - réservé, 49
- MS-DOS, 27
- nombre
 - à virgule, 7
 - décimal, 7
 - sexagésimal, 7
- numération
 - additive, 3
 - de position, 4
 - unaire, 3
- opérateur
 - d'adressage, 56
 - d'affectation, 73
 - d'indirection, 99
 - de contenu, 99
 - de pointeur, 98
 - de référence, 99
- ordinateur, v
- overflow*, 45
- paramètre
 - d'une fonction, 86
 - effectif, 86
 - formel, 86
- PASCAL, 25
- passage
 - par adresse, 99
- phase
 - d'implémentation, 78
 - de conception, 78
- point décimal, 37
- pointeur, 98
- POSIX, 110
- pow(), 41
- pragmatique, 76
- préprocesseur, 32
- printf(), 30
- progiciel, 25
- programmation, 24
- programme, 24
 - d'ordinateur, 15
 - objet, 25
 - principal, 83
 - source, 24
 - valide, 30
- programmeur, 24
- prompteur, 56
- prototype
 - complet, 95
 - de fonction, 94
 - simplifié, 95
- putch(), 102
- putchar(), 102
- putche(), 102
- raffinement
 - successif, 83, 100
- référence
 - directe, 98
- répétition, 66
- représentation

- décimale, 37
- scientifique, 38
- return**, 86
- réutilisation, 110
- saisie
 - au vol, 103
- sémantique, 33
 - attendue, 33
 - réelle, 33
- séquence d'échappement, 43
- séquencement, 62
- short**, 50
- signed**, 50
- sin()**, 41
- sinh()**, 41
- sous-programme, 83
- spécification, 78
- sqrt()**, 41
- stdio.h**, 31
- string*, 57
- structure
 - de contrôle, 61
- style
 - de programmation, 76
- syntaxe, 30
- table
 - d'addition, 5
 - de multiplication, 5
- tan()**, 41
- tanh()**, 41
- termios.h**, 110
- test, 70
- thèse, 80
- transmission
 - par référence, 91
 - par valeur, 91
- TURING, Alan, 24
- typage, 50
- type, 48, 50
 - constructeur, 52
- underflow*, 45
- Unix, 27
- unsigned**, 50
- valeur
 - de retour, 86
 - sentinelle, 67
- signal, 67
- variable, 48
 - de boucle, 67
 - de contrôle, 67
 - globale, 92
 - locale, 87, 93
 - pointeur, 98
- void**, 30, 85
- VON NEUMANN, John, 24
- warning*, 32
- Windows, 27
- xemacs**, 27
- zéro, 4
- ZUSE, Konrad (1910–1995), 16