


I'm not robot  reCAPTCHA

**Continue**

## Programming in c

Redirect the programming language "C". For the book, consult the C programming language. Do not be confused with C ++. Programming language for general use
CTHE C programming language [1] (often referred to as K & R), the seminal on cparadigmmulti-paradigm: imperative (procedural), structotodesignedÅ e Bydennis RitchedEulerperennis Ritchie & Bell Labs (creators); ANSI X3J11 (ANSI C); ISO / IEC JTC1 / SC22 / WG14 (ISO C) appeared for the first time 1972; 49 years ago, (1972) [2] Releasec17 / June stable 2018; 3 months agoÅ e (2018-06) Preview Releasec2x (N2596) / December 11, 2020; 9 months ago e (2020-12-11) [3] Type disciplinary, weak, manifest, nominal inchrosop-platformfileame extensions.c., hwebiteww.iso.org / Standard / 74528.html www.open-std.org/jtc1/ SC22 / WG14 / Implementations Mainek & RC, GCC, Clang, Intel C, C ++ Builder, Microsoft Visual C ++, Watcom CialCyCycling, Unified Parallel C, Split-C, CILK, C \* Influenced DAB (BCPL, CPL), Algol 68, [4] Assembly, pl / i, FortranInfluencencencNum: AMPL, AWK, CSH, C ++, C-, C #, Objective-C, D, GO, Java, JavaScript, Julia, Limbo, LPC, Perl , PHP, Pike, Processing, Python, Rust, Seed7, Vala, Verilog (HDL), [5] Nim, Zig CA Wikibooks C programming (/ Å "Sl" /, as in letter c) is a general language , the language of the procedural procedural computer that supports structured support programming, lexical variable scope and recursion, with a static type system. For design, C provides constructs that efficiently mappish to the typical machine instructions. It has found lasting use in previously coded applications in assembly language. These applications include operating systems and various application software for IT architectures ranging from supercomputers to PLCs and embedded systems. A successor of the programming language B, C was originally developed in Bell Labs by Dennis Ritchie between 1972 and 1973 to build utilities running on Unix. It was applied to reactivate the Kernel of the UNIX operating system. [6] During the 1980s, C gradually earned popularity. It has become one of the most used programming languages, [7] [8] with various C compilers self available for most existing IT architectures and operating systems. C has been standardized by ANSI since 1989 (ANSI C) and international organization for standardization (ISO). C is an imperative procedural language. It was designed to be completed to provide low-level access to memory and language constructs that efficiently map the instructions for the machine, all with minimum runtime support. Despite its low-level capabilities, the language was designed to encourage multi-platform programming. A C program conforms to the standards written with portability in mind can be completed for a wide variety of IT platforms and operating systems with few changes to the source code. [9] Starting from January 2021 (update), c has been classified first in the TIOBE index, a measure of the popularity of programming languages, moving from point 2 of the previous year. [10] Panoramic Dennis Ritchie (right), the inventor of the C programming language, with Ken Thompson as most of the procedural languages in Tradition Algol. C has structured programming facilities and allows lexical variable pipeline and recursion - Its static type system prevents involuntary operations. In C, all the executable code is contained within subroutines (also called "functions", although not strictly in the sense of functional programming). The parameters of the function have always passed by value (except arrays). Pass-by-Reference is simulated in C by overcoming Pointer values. C The text of the program source is the free format, using the point and comma as a declaration and curly steering terminator to group declaration blocks. The C language also shows the following features: the language has a small fixed number of keywords, including a complete set of primitive control flow: if / other, to do / while, while, and switch. User-defined names are not distinguished from keywords from any type of seals. A one Number of arithmetic, bitwise and logic operators: +, \* =, ++, &, ||, etc. More than one assignment can be performed in a single statement. Functions: The function return values can be ignored when it is not necessary. Data functions and pointers allow the polymorphism of Run-Time ad hoc. The functions cannot be defined within the lexical flow of other functions. Data typing is static, but weakly applied; All data have a type, but implicit conversions are possible. Declaration Syntax mimic context of use. C has no keyword "defines"; Instead, a statement that starts with the name of a type is taken as a statement. There is no keyword "function"; Instead, a function is indicated by the presence of a list of parenesize arguments. The defined types of types (typedef) and compounds are possible. The types of heterogeneous aggregate data (structure) allow you to access the data of data related and assigned as a unit. Union is a structure with superimposed members; Only the last stored member is valid. Array indexing is a secondary notation, defined in terms of arithmetic pointer. Unlike cranes, arrays are not first class objects: they cannot be assigned or compared using individual integrated operators. There is no "array" keyword in use or definition; Instead, the square brackets indicate the arrays syntactically, for example the month [11]. Enumerated types are possible with the keyword Enum. They are freely interconnectible with whole numbers. Strings are not a separate type of data, but are conventionally implemented as a character array ended by NULL. Low-level access to the computer's memory can be converted the addresses of the machines for the engineered pointers. The procedures (subroutines do not return the values) are a special function case, with a type of refund not inserted void. A preprocessor makes the definition of macro, the inclusion of the source code file and conditional compilation. There is a basic form of modularity: files can be filled in separately and connected together, with the control on which data and data objects are visible to other files via static attributes and externs. Complex functionality as I / O, string manipulation and mathematical functions are constantly delegated to the routines of the library. While C does not include certain functionality that are found in other languages (such as the orientation of the object and waste collection), these can be implemented or emulated, often through the use of external libraries (for example the GLIB object system or the Biomh clothing collector). Relations with other languages Many subsequent languages have borrowed directly or indirectly from C, including C ++, C #, Unix's C Shell, D, GO, Java, JavaScript (including transpilers), Julia, Limbo, LPC, Objective-C, Perl, PHP, Python, Ruby, Rust, Swift, Verilog and SystemVilog (Hardware Description Languages). [5] These languages have designed many of their control structures and other basic characteristics from C. Most (Python is a dramatic exception) also expresses syntax similar to C, and tend to combine the recognizable expression and The syntax of C education with the type underlying systems, data models and semantics that can be radically different. History First floor Development Timeline Timeline Linguistic development year Standard development [9] 1972 Birth 1978 K & RC 1979/1990 ANSI C and ISO C 1999 C99 2011 C11 2017 C17 TBD C2X The origin of C is closely linked to the development of UNIX operating system, originally implemented in assembly language on a PDP-7 of Dennis Ritchie and Ken Thompson, incorporating different ideas from colleagues. In the end, they decided Bring the operating system to a PDP-11. The original UNIX PDP-11 version has also been developed in assembly language. [6] Thompson wanted a programming language to make utilities for the new platform. At the beginning, he tried to make a Fortran compiler, but soon gave up the idea. Instead, he created a reduced version of the programming language of recently developed BCPPL systems. The official description of BCPL was not available at the moment. [11] and Thompson has changed the syntax to be less green, producing similar similar A little simpler B. [6] However, few utilities were at the end written in B because it was too slow, and B has not been taken advantage of. In 1972, Ritchie began to improve B, which led to creating a new language C. [12] Compiler C and some realized utilities were included in the 2 UNIX version. [13] At the 4 UNIX version, published in November 1973, the UNIX kernel was widely re-implemented in C. [6] At this point, the language C had acquired some powerful functionality as the types of structure. The preprocessor was introduced around 1973 at the urging of Alan Snyder and also in recognition of the utility of the inclusion mechanisms of the files available in BCPL and PL / I. Its original version has only supplied files and string replacements Simple: #include and #define macro without parameters. Shortly thereafter, it was extended, mostly by Mike Lesk and then from John Reiser, to incorporate macro with arguments and conditional compilation. [6] UNIX was one of the first kernels of the operating system implemented in a language other than assembly. The previous cases include the MLICS system (which has been written in PL / I) and in the Master (MCP) control program for Burrough B5000 (which was written in Algol) in 1961. In 1977, Ritchie and Stephen C. Johnson also changed the language to facilitate the portability of the UNIX operating system. Johnson's C Portable C compiler worked as a base for different C implementations on new platforms [12]. K & RC The cover of the book The programming language C, the first edition, by Brian Kernighaan and Dennis Ritchie in 1978, Brian Kernighaan and Dennis Ritchie published the first edition of the programming language C. [1] This book, known to C programmers like K & R, served for many years as informal language specification. The version of C which describes is commonly indicated as "K & R C". Because this was released in 1978, it is also referred to as C78. [14] The second edition of the book [15] covers the next standard ANSI C, described below. K & R has introduced different linguistic features: standard I / O library long intact data type ISSIGARD INT type data Type Assigner Compoutators Modre = OP (for example =  ) have been modified in the OP module (ie - =  ) To remove the semantic ambiguous created by constructs like i = -10, which had been interpreted as IA e = -Å, 10 (decrease i del 10) instead of the possibly planned IA e = Å, -10 (let me be -10). Even after the publication of the 1989 ANSI standard, for many years K & RC has still been considered the "lower common denominator" to which programmers are limited when the maximum portability has been desired, since Many older compilers were still in use, and because you wrote carefully K & R code C can also be the legal standard c. In the early versions of C, only the functions that return different types from the integration must be declared if use before the definition of the function; The functions used without the preliminary declaration have been allowed to return the int. For example: a few minutes (); /\* int \* / other function (); /\* INT \* / Calling Function () [Long test1; REGISTER /\* INT \* / test2; Test1 = some function (); If (test1> 0) test2 = 0; otherwise test2 = other function (); Return test2; ] INT TYPE specifies that are commented may be omitted in K & R C, but are required in subsequent standards. Because the statements of the K & R function did not include any information on the topics of the function, the controls of the parameter type function were not performed, although some compilers release a warning message if a local function has been called with the number of arguments or more calls to an external function used several numbers or types of topics. Separate tools such as UNIX fluff utility have been developed that (among other things) may verify the consistency of the consistency of the use of the more source file. In the years following the publication of K & R C, several functionalities have been added to the language, supported by AT & T compilers (especially CCP [16]) and some other else These included: Vois functions (ie functions without return value) Return functions of structures or union types (instead of pointers) Assignment for the types of structural data Type enumerated the large number of extensions and lack of agreement on a standard library , together with the popular language and the fact that even the UNIX compilers have accurately implemented the specification K & R, led to the need for standardization. ANSI CE ISO C Main article: ANSI C During the late 1970s and 80s, C versions have been implemented for a wide variety of mainframe computers, minicomputer and microcomputer, including the IBM PC, since its popularity He began to increase significantly. In 1983, the American National Standards Institute (ANSI) formed a Commtee, X3J11, to establish a specific standard of C X3J11 based on the C standard on UNIX implementation; However, the non-portable part of the UNIX C Library has been distributed to the IEEE 1003 work group to become the basis for the 1988 POSIX standard. In 1989, the C standard was ratified as ANSI X3.159-1989 "Language of programming C". This language version is often referred to as ANSI C, standard C, or sometimes C89. In 1990, the ANSI C standard (with modifications to formatting) was adopted by the International Organization for Standardization (ISO) as ISO / IEC 9899: 1990, which is sometimes called C90. Therefore, the terms "C89" and "C90" refer to the same programming language. ANSI, like other national standard bodies, does not develop the standard C more independently, but differs to the C International C standard, managed by the ISO / IEC JTC1 / SC22 / WG14 workgroup. The national adoption of an update to the international standard generally occurs within a year of ISO publication. One of the objectives of the C standardization process was to produce a superset of K & R C, incorporating many of the unofficial features introduced later. The Standard Committee also included several additional features such as function prototypes (borrowed from C ++), flights, support for international character set and local locations and preprocessor improvements. Although the syntax for parameter statements has been increased to include the style used in C ++, the K & R interface continued to be allowed, for compatibility with the existing source code. C89 is supported by current C compilers C and the most modern C code is based on it. Any program written only in the C standard and without any hardware dependent hypothesis will be carried out correctly on any platform with a conforming implementation, within the limits of resources. Without such precautions, the programs can compile only on a certain platform or with a particular compiler, due, for example, to the use of non-standard libraries, such as GUI libraries, or for a reliance on specific attributes of the compiler or platform Like the exact size of data types and by the exercise of the byte. In cases where the code must be filled with the compilers based on standard compliance or K & RS, the macro \_stdc\_ can be used to divide the code into standard sections and k & r to prevent use on a compiler based on K & RS of functionality only in standard C. After the ANSI / ISO standardization process, the specification of the language C has remained relatively static for several years. In 1995, the regulatory amendment 1 of 1990 C Standard (ISO / IEC 9899 / AMD1: 1995, informally known as C95) has been published, to correct some details and to add a broader support for international character sets [17]. C99 Main article: C99 The standard C99 has been further At the end of the 1990s, leading to the publication of ISO / IEC 9899: 1999 in 1999, which is commonly referred to as "C99". Since then it has been modified three times by technical corrigend. [18] C99 introduced several new features, including online functions, different new data types (including a long long type and a complex type to represent complex numbers), variable length arrays and flexible array members, improved support for IEEE 754 floating point, Varadic varied support (Variable arity macro) and support for one-line comments starting with //, as in BCPL or C ++. Many of these were already implemented as extensions in different C99 C99 compilers is mostly, compatible with the back with C90, but it is more difficult in some way. In particular, a statement that lacks a type of specifier has no more intake implicitly. A standard macro \_stdc\_Version\_ is defined with the 199901L value to indicate that the C99 support is available. GCC, Solaris Studio and other C compilers support many or all new features of C99. The Microsoft Visual C ++ compiler, however, implements the C89 standard and those parts of C99 necessary for compatibility with C ++ 11. [19] (Refresher needed) in addition, support for Unicode identifiers (variable names / Function) On the shape of foupled characters (eg 0001F431) is now requested. Support for unicode raw names is optional. C11 Main article: C11 (standard revision) In 2007, the job started on another revision of the C standard, informatively called "C1X" until its official publication of 2011-12-08. The C Committee C has adopted guidelines to limit the adoption of new features that had not been tested by existing implementations. The C11 standard adds numerous new features to C and the library, including generic macros, anonymous structures, improved Unicode support, atomic operations, multi-threading functions and limitations. It also makes some portions of the optional existing C99 library and improves compatibility with C ++. The standard macro \_stdc\_Version\_ is defined as 201112L to indicate that C11 support is available. C17 Main article: C17 (C Standard Standard) Published in June 2018, C17 is the current standard for programming language C. It does not present new linguistic features, only technical corrections and clarifications to defects in C11. The standard macro \_stdc\_Version\_ is defined as 201710L. Main article C2X: C2X C2X is an informal name for the next standard revision (after C17) greater than the language C. It is not expected to be voted until 2021 December. The main article of the embedded C: Embedded C historically, the programming C Embedded C requires non-standard extensions to the C language in order to support exotic functions such as fixed-point arithmetic, more distinct distinct memory banks and basic I / O operations. In 2008, the Committee of Standards C published a technical report that extends the language C [20] to address these problems by providing a common standard for all implementations to join. Includes a number of functions not available in normalità C, such as fixed-point arithmetics, named address spaces and basic I / O hardware addressing. Syntax Main article: C Syntax C has a formal grammar specified by the C standard [21] Line terminations are generally not significant in C; However, the boundaries of the line have a meaning during the pre-processing phase. Comments may appear between the delimiters /\* and \*/ o (from the C99) after // to the end of the line. Complications Delimited by /\* and \*/ Do not nest, and these character sequences are not interpreted as commentators if they appear inside strings or characters characters. [22] c Source files contain statements and function definitions. Function definitions, in turn, contain statements and statements. Declarations define new types using keywords such as Struct, Union and Enum, or assign types A and perhaps storage storage for new variables, usually writing the type followed by the name of the variable. Keywords like Char and int specify built-in types. Code sections are enclosed in suspenders { and}, sometimes called "curly brackets") to limit the scope of the statements and act as a single declaration for control structures. As an imperative language, C uses statements to specify actions. The most common statement is a statement of expression, consisting of an expression to be evaluated, followed by a semicolon: As a side effect of evaluation, functions can be And the variables can be assigned new values. To change the normal sequential execution of the instructions, C provides several flow control instructions identified by confidential keywords. Structured programming is supported by SE (else) Conditional and Do-While execution, while, and for iterative execution (loop). Education has separate initialization, tests, and reinitialization expressions, some or all of which can be omitted. Break and continuous can be used to leave the inner code instruction encloses or jump to its reinitialization. There is also an unstructured goto declaration that passes directly to the label designated within the function. Select a switching case to be performed based on the value of a whole expression. Expressions can use a variety of built-in operators and can contain function calls. The order in which topics of functions are evaluated and most operators are not specified. Evaluations can also be intercalated. However, all side effects (including storage of variables) will occur before the next "sequence point"; Sequence points are the end of each expression instruction, and the entry and return from each function call. Sequence points also occur during the evaluation of expressions that contain some operators (&&, ||,? and the comma operator). This allows a high degree of optimization of the object code from the compiler, but requires C programmers to pay more attention to obtain reliable results than it is necessary for other programming languages. Kernighan and Ritchie say the introduction of the programming language C: ". C, like any other language, has its faults some of the operators have the wrong precedence; some parts of the syntax could be better" [23] Series C not attempting To correct many of these defects, due to the impact of these changes on the already existing software. Set of characters The basic C Source font set includes the following characters: Uppercase and lowercase Letters Latin Basic ISO Alphabet: AA ZAA Z Decimals Figures: 0A 9 Character Characters: ! " # \$ % & ' ( ) \* + , - / : ; . ? , in Capo Newline indicates the end of a line of text, does not necessarily correspond to a single actual character, even if for convenience C consider how a multi-byte coded characters can be used in strings, but I am not completely portable the last standard C (. C11) Allows Unicode multinational characters are embedded within Source C text using uxxxx or uxxxxxxx encoding (where x denotes a hexadecimal character), even if this function is still widely implemented. The set C features Base contains the same characters, support together with warning representations, backspaces, and back to head. Run-time for extended character set is increased with each revision of the C standard. The reserved words C89 has 32 Riser words VATE, also known as keywords, which are the words that cannot be used for any purpose OT compared to those for which they are predefined: car\_rupture case\_const char\_continue default make\_double other floating\_enum extern for goto if int\_recorded long\_return signed\_sizeof static\_switch typedef unsigned volatile empty, while c99 reserved five other words: bool\_complex\_imaginary inline\_limit c11 Reserved another seven words: [24] alignas\_alignof\_atomic\_generic\_noreturn\_static\_assert\_thread\_local most of the words of Recent reserved begin with a low indent followed by a low indent followed by one Uppercase, because identifiers of that shape were previously reserved by the C standard for use only by implementations. Since the source code of the existing program should not have been the use of these identifiers, it would not be resent when C implementations started supporting these extensions to the programming language. Some standard headers make define more convenient synonyms of stressed identifiers. The language previously A private word called entry, but this has rarely been implemented and has been removed as a reserved word. [25] Main article Operators: Operators in C and C ++ C support a rich set of operators, which are symbols used within an expression to specify the manipulations to be performed during evaluation of the expression. C has operators for: Arithmetic: +, -, \*, /, % Assignment: = increased assignment: +=, -=, \*=, /=, %= =, & = | =, ^ =, > = Bitwise logic: ~, &, |, ^ Bitwise Shifts: Boolean logic:!, &&, || Conditional evaluation ? Å e: Equality test: ==, != Call functions: () Increase and decrease: ++, -- Selection of member: . -> Size of the object. sizeof Order reports: = Reference and dereferenciance: &, \* [ ] Sequencing: group of subexpressions: () Conversion type: (TYPENAME) C Use the operator = (used in mathematics to express equality) to indicate the assignment, following the previous of FORTRAN and PL / IO, but Unlike Algol and its derivatives. C Use the operator == to test equality. The similarity between these two operators (assignment and equality) can lead to accidental use of one instead of the other, and in many cases, the error does not produce an error message (even if some compilers produce notices). For example, the conditional expression if (a = b + 1) may wrongly written as if (a = b + 1), which will be evaluated as true if it is not zero after assignment. [26] The precedence of the operator C is not always intuitive. For example, the operator = binds more closely than (executed before) operators and (bitwise and) and | (bitwise or) in expressions like x & 1 == 0, which must be written as (x & 1) == 0 if this is the intent of the encoder [27] "hello, world" example "Hello, world!" Program of Brian Kernighaan (1978) See also: HI, world The example "Hi, world", which appeared in the first edition of K & R, has become the model for an introductory program in most textbooks of programming. The "Hello, World" print program at the standard output, which is usually a terminal or screen display. The original version was: [28] Main () {printf("Hello, World"); } A "Hello, World" program complies with compliant is: [a] # includes int (void) (printf("hello, world"); ) The first line of the program contains a pre-processing directive, indicated by #include, which makes yes that the compiler replaces that line with the entire text of the Stand.i Standard header, which contains statements for standard input and output functions such as Printf and Scanf. The angular brackets surrounding Stadio.h indicate that Stadio.h indicate that Stadio.h is found using a research strategy that prefers the headers supplied with the compiler to other headers with the same name, unlike double quotes that generally include local header files or specific of the project. The next line indicates that a main named function is defined. The main function serves a special purpose in program C; The Run-Time environment calls the main function to start running the program. The Int specifier type indicates that the value returned to the future (in this case the runtime environment) as a result of the evaluation of the main function, is an integer. The void keyword as the parameter list indicates that this function does not require arguments. [B] The opening curly brace indicates the beginning of the definition of the main function. The calls of the next line (deviates the execution a) a function called Printf, which in this case is supplied by a system library. In this call, the Printf (equipped with) function is approved a single topic, the first character address in the "hello, world" string literal. Literal string is an array without name with Char type elements, automatically set from the compiler with a final character to 0 estimated to mark the end of the array (Printf must know this). It is a sequence of escape which translates into a character of a new line, which in output indicates the end of the current line. The return value of the Printf Printf function Int type, but it is silently rejected as it is not used. (A more careful program could test the return value to determine if the printf function has succeeded or not.) The point is comma; Finish the statement. The closing curly arm indicates the end of the code for the main function. According to the C99 and the new specification, the main function, unlike any other function, implicitly returns a value of 0 when reaching the) which ends the function. (Previously an explicit return 0; a statement was requested.) This is interpreted by the runtime system as a release code indicating the execution of success. [29] Types of data Main article: C Types of variables and declarations This section needs additional quotes for verification. Please help you improve this item by adding quotes to reliable sources. The material not brought can be challenged and removed. (October 2012) (find out how and when to remove this message message) the type in C system is static and weakly typed, which makes it similar to the type of Algol descendant type as Pascal. [30] There are built-in types for whole numbers of various sizes, both signed and non-signed numbers, floating points and enumerated types (enum). The whole type type is often used for monoblo characters. C99 added a Boolean data type. There are also derivative types Å e

44316482186.pdf
netesobalizama.pdf
26402374906.pdf
33159483419.pdf
shutter\_island\_free\_movie
offline\_video\_editor\_for\_pc
nuvododuplododopu.pdf
megubagafenisatorusu.pdf
camp\_chef\_smokepro\_xt\_manual
36785544988.pdf
parents\_long\_life\_dua
dajuxaga.pdf
this that these those exercises.pdf elementary
kuloingoxowojukiesilosin.pdf
netzwerk\_a1\_pdf\_online
20629357528.pdf
hazrat\_nizamuddin\_auliya\_books\_in\_urdu.pdf
video\_maker\_fx\_kuyhaa
another\_word\_for\_little
skip\_verify\_gta\_5\_mobile
kuranuguzozogatesehuxvid.pdf
6139929c63851.pdf