Basic

I'm not robot!

Please enable JavaScript to view this page. Family of programming languages For other uses, see BASIC (disambiguation). BASICAtari BASIC (1979) for the Atari 8-bit familyParadigmNon-structured, later procedural, later object-orientedDesigned by John G. Kemeny Thomas E. Kurtz First appearedMay 1, 1964; 58 years ago (1964-05-01)Major implementations Dartmouth BASIC Tiny BASIC Applesoft BASIC Atari BASIC Sinclair BASIC Commodore BASIC BBC BASIC TI-BASIC Casio BASIC Microsoft BASIC QB64 FreeBASIC Liberty BASIC PowerBASIC QuickBASIC Influenced by ALGOL 60 FORTRAN II JOSS Influenced COMAL Visual Basic Visual Basic .NET GRASS BASIC Programming at Wikibooks BASIC (Beginners' All-purpose Symbolic Instruction Code)[1] is a family of general-purpose, high-level programming languages designed for ease of use. The original version was created by John G. Kemeny and Thomas E. Kurtz at Dartmouth College in 1964. They wanted to enable students in non-scientific fields to use computers. At the time, nearly all computers required writing custom software, which only scientists and mathematicians tended to learn. In addition to the program language, Kemeny and Kurtz developed the Dartmouth Time Sharing System (DTSS), which allowed multiple users to edit and run BASIC programs simultaneously on remote terminals. This general model became very popular on minicomputer systems like the PDP-11 and Data General Nova in the late 1960s and early 1970s. Hewlett-Packard produced an entire computer line for this method of operation, introducing the HP2000 series in the late 1960s and continuing sales into the 1980s. Many early video games trace their history to one of these versions of BASIC. The emergence of microcomputers in the mid-1970s led to the development of multiple BASIC dialects, including Microsoft BASIC in 1975. Due to the tiny main memory available on these machines, often 4 KB, a variety of Tiny BASIC dialects was also created. BASIC was available for almost any system of the era, and became the de facto programming language for home computer systems that emerged in the late 1970s. These PCs almost always had a BASIC interpreter installed by default, often in the machine's firmware or sometimes on a ROM cartridge. BASIC declined in popularity in the 1990s, as more powerful microcomputers came to market and programming languages with advanced features (such as Pascal and C) became tenable. In 1991, Microsoft released Visual Basic, combining an updated version of BASIC with a visual forms builder. This reignited use of the language and "VB" remains a major programming language[citation needed] in the forms of VBA and VB.NET. Origin John G. Kemeny was the math department chairman at Dartmouth College. Based largely on his reputation as an innovator in math teaching, in 1959 the school won an Alfred P. Sloan Foundation award for $500,000 to build a new department building.[2] Thomas E. Kurtz had joined the department in 1956, and from the 1960s Kemeny and Kurtz agreed on the need for programming literacy among students outside the traditional STEM fields. Kemeny later noted that "Our vision was that every student on campus should have access to a computer, and any faculty member should be able to use a computer in the classroom whenever appropriate. It was as simple as that."[3] Kemeny and Kurtz had made two previous experiments with simplified languages, DARSIMCO (Dartmouth Simplified Code) and DOPE (Dartmouth Oversimplified Programming Experiment). These did not progress past a single freshman class. New experiments using Fortran and ALGOL followed, but Kurtz concluded these languages were too tricky for what they desired. As Kurtz noted, Fortran had numerous oddly-formed commands, notably an "almost impossible-to-memorize convention for specifying a loop: DO 100, I = 1, 10, 2. Is it '1, 10, 2' or '1, 2, 10', and is the comma after the line number required or not?"[3] Moreover, the lack of any sort of immediate feedback was a key problem; the machines of the era used batch processing and took a long time to complete a run of a program. While Kurtz was visiting MIT, John McCarthy suggested that time-sharing offered a solution; a single machine could divide up its processing time among many users, giving them the illusion of having a (slow) computer to themselves.[4] Small programs would return results in a few seconds. This led to increasing interest in a system using time-sharing and a new language specifically for use by non-STEM students.[3] Kemeny wrote the first version of BASIC. The acronym BASIC comes from the name of an unpublished paper by Thomas Kurtz.[5] The new language was heavily patterned on FORTRAN II; statements were one-to-a-line, numbers were used to indicate the target of loops and branches, and many of the commands were similar or identical to Fortran. However, the syntax was changed wherever it could be improved. For instance, the difficult to remember DO loop was replaced by the much easier to remember FOR I = 1 TO 10 STEP 2, and the line number used in the DO was instead indicated by the NEXT I.[a] Likewise, the cryptic IF statement of Fortran, whose syntax matched a particular instruction of the machine on which it was originally written, became the simpler IF I=5 THEN GOTO 100. These changes made the language much less idiosyncratic while still having an overall structure and feel similar to the original FORTRAN.[3] The project received a $300,000 grant from the National Science Foundation, which was used to purchase a GE-225 computer for processing, and a Datanet-30 realtime processor to handle the Teletype Model 33 teleprinters used for input and output. A team of a dozen undergraduates worked on the project for about a year, writing both the DTSS system and the BASIC compiler.[3] The first version BASIC language was released on 1 May 1964.[6][7] Initially, BASIC concentrated on supporting straightforward mathematical work, with matrix arithmetic support from its initial implementation as a batch language, and character string functionality being added by 1965. Usage in the university rapidly expanded, requiring the main CPU to be replaced by a GE-235,[3] and still later by a GE-635. By the early 1970s there were hundreds of terminals connected to the machines at Dartmouth, some of them remotely. Wanting use of the language to become widespread, its designers made the compiler available free of charge. In the 1960s, software became a chargeable commodity; until then, it was provided without charge as a service with the very expensive computers, usually available only to lease. They also made it available to high schools in the Hanover, New Hampshire area and regionally throughout New England on Teletype Model 33 and Model 35 teleprinter terminals connected to Dartmouth via dial-up phone lines, and they put considerable effort into promoting the language. In the following years, as other dialects of BASIC appeared, Kemeny and Kurtz's original BASIC dialect became known as Dartmouth BASIC. New Hampshire recognized the accomplishment in 2019 when it erected a highway historical marker in Hanover describing the creation of "the first user-friendly programming language".[8] Spread on time-sharing services The emergence of BASIC took place as part of a wider movement towards time-sharing systems. First conceptualized during the late 1950s, the idea became so dominant in the computer industry by the early 1960s that its proponents were speaking of a future in which users would "buy time on the computer much the same way that the average household buys power and water from utility companies".[9] General Electric, having worked on the Dartmouth project, wrote their own underlying operating system and launched an online time-sharing system known as Mark I. It featured BASIC as one of its primary selling points. Other companies in the emerging field quickly followed suit; Tymshare introduced SUPER BASIC in 1968, CompuServe had a version on the DEC-10 at their launch in 1969, and by the early 1970s BASIC was largely universal on general-purpose mainframe computers. Even IBM eventually joined the club with the introduction of VS-BASIC in 1973.[10] Although time-sharing services with BASIC were successful for a time, the widespread success predicted earlier was not to be. The emergence of minicomputers during the same period, and especially low-cost microcomputers in the mid-1970s, allowed anyone to purchase and run their own systems rather than buy online time which was typically billed at dollars per minute.[b][11] Spread on minicomputers The HP 2000 system was designed to run time-shared BASIC as its primary task. BASIC, by its very nature of being small, was naturally suited to porting to the minicomputer market, which was emerging at the same time as the time-sharing services. These machines had very small main memory, perhaps as little as 4 KB in modern terminology,[c] and lacked high-performance storage like hard drives that make compilers practical. On these systems, BASIC was normally implemented as an interpreter rather than a compiler due to the reduced need for working memory.[d] A particularly important example was HP Time-Shared BASIC, which, like the original Dartmouth system, used two computers working together to implement a time-sharing system. The first, a low-end machine in the HP 2100 series, was used to control user input and save and load their programs to tape or disk. The other, a high-end version of the same underlying machine, ran the programs and generated output. For a cost of about $100,000, one could own a machine capable of running between 16 and 32 users at the same time.[12] The system, bundled as the HP 2000, was the first mini platform to offer time-sharing and was an immediate runaway success, catapulting HP to become the third-largest vendor in the minicomputer space, behind DEC and Data General (DG).[13] DEC, the leader in the minicomputer space since the mid-1960s, had initially ignored BASIC. This was due to their work with RAND Corporation, who had purchased a PDP-6 to run their JOSS language, which was conceptually very similar to BASIC.[14] This led DEC to introduce a smaller, cleaned up version of JOSS known as FOCAL, which they heavily promoted in the late 1960s. However, with timesharing systems widely offering BASIC, and all of their competition in the minicomputer space doing the same, DEC's customers were clamoring for BASIC. After management repeatedly ignored their pleas, David H. Ahl took it upon himself to buy a BASIC for the PDP-8, which was a major success in the education market. By the early 1970s, FOCAL and JOSS had been forgotten and BASIC had become almost universal in the minicomputer market.[15] DEC would go on to introduce their updated version, BASIC-PLUS, for use on the RSTS/E time-sharing operating system. During this period a number of simple text-based games were written in BASIC, most notably Mike Mayfield's Star Trek. David Ahl collected these, some ported from FOCAL, and published them in an educational newsletter he compiled. He later collected a number of these into book form, 101 BASIC Computer Games, published in 1973.[16] During the same period, Ahl was involved in the creation of a small computer for education use, an early personal computer. When management refused to support the concept, Ahl left DEC in 1974 to found the seminal computer magazine, Creative Computing. The book remained popular, and was re-published on several occasions.[17] Explosive growth: the home computer era See also: List of computers with on-board BASIC Commodore BASIC v2.0 on the Commodore 64 MSX BASIC version 3.0 "Train Basic every day!"—reads a poster (bottom center) in a Russian school (c. 1985–1986) The introduction of the first microcomputers in the mid-1970s was the start of explosive growth for BASIC. It had the advantage that it was fairly well known to the young designers and computer hobbyists who took an interest in microcomputers, many of whom had seen BASIC on minis or mainframes. Despite Dijkstra's famous judgement in 1975, "It is practically impossible to teach good programming to students that have had a prior exposure to BASIC: as potential programmers they are mentally mutilated beyond hope of regeneration",[18] BASIC was one of the few languages that was both high-level enough to be usable by those without training and small enough to fit into the microcomputers of the day, making it the de facto standard programming language on early microcomputers. The first microcomputer version of BASIC was co-written by Bill Gates, Paul Allen and Monte Davidoff for their newly formed company, Micro-Soft. This was released by MITS in punch tape format for the Altair 8800 shortly after the machine itself,[19] immediately cementing BASIC as the primary language of early microcomputers. Members of the Homebrew Computer Club began circulating copies of the program, causing Gates to write his Open Letter to Hobbyists, complaining about this early example of software piracy. Partially in response to Gates's letter, and partially to make an even smaller BASIC that would run usefully on 4 KB machines,[e] Bob Albrecht urged Dennis Allison to write their own variation of the language. How to design and implement a stripped-down version of an interpreter for the BASIC language was covered in articles by Allison in the first three quarterly issues of the People's Computer Company newsletter published in 1975 and implementations with source code published in Dr. Dobb's Journal of Tiny BASIC Calisthenics & Orthodontia: Running Light Without Overbyte. This led to a wide variety of Tiny BASICs with added features or other improvements, with versions from Tom Pittman and Li-Chen Wang becoming particularly well known.[20] Micro-Soft, by this time Microsoft, ported their interpreter for the MOS 6502, which quickly become one of the most popular microprocessors of the 8-bit era. When new microcomputers began to appear, notably the "1977 trinity" of the TRS-80, Commodore PET and Apple II, they either included a version of the MS code, or quickly introduced new models with it. By 1978, MS BASIC was a de facto standard and practically every home computer of the 1980s included it in ROM. Upon boot, a BASIC interpreter in direct mode was presented. Commodore Business Machines included Commodore BASIC, based on Microsoft BASIC. The Apple II and TRS-80 each had two versions of BASIC, a smaller introductory version introduced with the initial releases of the machines and an MS-based version introduced as interest in the platforms increased. As new companies entered the field, additional versions were added that subtly changed the BASIC family. The Atari 8-bit family had its own Atari BASIC that was modified in order to fit on an 8 KB ROM cartridge. Sinclair BASIC was introduced in 1980 with the Sinclair ZX80, and was later extended for the Sinclair ZX81 and the Sinclair ZX Spectrum. The BBC published BBC BASIC, developed by Acorn Computers Ltd, incorporating many extra structured programming keywords and advanced floating-point operation features. As the popularity of BASIC grew in this period, computer magazines published complete source code in BASIC for video games, utilities, and other programs. Given BASIC's straightforward nature, it was a simple matter to type in the code from the magazine and execute the program. Different magazines were published featuring programs for specific computers, though some BASIC programs were considered universal and could be used in machines running any variant of BASIC (sometimes with minor adaptations). Many books of type-in programs were also available, and in particular, Ahl published versions of the original 101 BASIC games converted into the Microsoft dialect and published it from Creative Computing as BASIC Computer Games. This book, and its sequels, provided hundreds of ready-to-go programs that could be easily converted to practically any BASIC-running platform.[16][21][22] The book reached the stores in 1978, just as the home computer market was starting off, and it became the first million-selling computer book. Later packages, such as Learn to Program BASIC would also have gaming as an introductory focus. On the business-focused CP/M computers which soon became widespread in small business environments, Microsoft BASIC (MBASIC) was one of the leading applications.[23] In 1978, David Lien published the first edition of The BASIC Handbook: An Encyclopedia of the BASIC Computer Language, documenting keywords across over 78 different computers. By 1981, the second edition documented keywords from over 250 different computers, showcasing the explosive growth of the microcomputer era.[24] IBM PC and compatibles IBM Cassette BASIC 1.10 When IBM was designing the IBM PC, they followed the paradigm of existing home computers in wanting to have a built-in BASIC. They sourced this from Microsoft – IBM Cassette BASIC – but Microsoft also produced several other versions of BASIC for MS-DOS/PC DOS including IBM Disk BASIC (BASIC D), IBM BASICA (BASIC A), GW-BASIC (a BASICA-compatible version that did not need IBM's ROM) and QBasic, all typically bundled with the machine. In addition they produced the Microsoft BASIC Compiler aimed at professional programmers. Turbo Pascal-publisher Borland published Turbo Basic 1.0 in 1985 (successor versions are still being marketed by the original author under the name PowerBASIC). Microsoft wrote the windowed AmigaBASIC that was supplied with version 1.1 of the pre-emptive multitasking GUI Amiga computers (late 1985 / early 1986), although the product unusually did not bear any Microsoft marks. These later variations introduced many extensions, such as improved string manipulation and graphics support, access to the file system and additional data types. More important were the facilities for structured programming, including additional control structures and proper subroutines supporting local variables. However, by the latter half of the 1980s, users were increasingly using pre-made applications written by others rather than learning programming themselves; while professional programmers now had a wide range of more advanced languages available on small computers. C and later C++ became the languages of choice for professional "shrink wrap" application development.[25][26] Visual Basic Main article: Visual Basic (classic) In 1991, Microsoft introduced Visual Basic, an evolutionary development of QuickBASIC. It included constructs from that language such as block-structured control statements, parameterized subroutines and optional static typing as well as object-oriented constructs from other languages such as "With" and "For Each". The language retained some compatibility with its predecessors, such as the Dim keyword for declarations, "Gosub"/Return statements and optional line numbers which could be used to locate errors. An important driver for the development of Visual Basic was as the new macro language for Microsoft Excel, a spreadsheet program. To the surprise of many at Microsoft who still initially marketed it as a language for hobbyists, the language came into widespread use for small custom business applications shortly after the release of VB version 3.0, which is widely considered the first relatively stable version. While many advanced programmers still scoffed at its use, VB met the needs of small businesses efficiently as by that time, computers running Windows 3.1 had become fast enough that many business-related processes could be completed "in the blink of an eye" even using a "slow" language, as long as large amounts of data were not involved. Many small business owners found they could create their own small, yet useful applications in a few evenings to meet their own specialized needs. Eventually, during the lengthy lifetime of VB3, knowledge of Visual Basic had become a marketable job skill. Microsoft also produced VBScript in 1996 and Visual Basic .NET in 2001. The latter has essentially the same power as C# and Java but with syntax that reflects the original Basic language. The IDE, with its event-driven GUI builder, was also influential on other tools, most notably Borland Software's Delphi for Object Pascal and its own descendants such as Lazarus.[27] Mainstream support for the final version 6.0 of the original Visual Basic ended on March 31, 2005, followed by extended support in March 2008.[28] On March 11, 2020, Microsoft announced that evolution of the VB.NET language had also concluded, although it was still supported.[29] Meanwhile, competitors exist such as Xojo and Gambas. Three modern Basic variants: Mono Basic, OpenOffice.org Basic and Gambas Post-1990 versions and dialects Many other BASIC dialects have also sprung up since 1990, including the open source QB64 and FreeBASIC, inspired by QBasic, and the Visual Basic-styled RapidQ, Basic For Qt and Gambas. Modern commercial incarnations include PureBasic, PowerBASIC, Xojo, Monkey X and True BASIC (the direct successor to Dartmouth BASIC from a company controlled by Kurtz). Several web-based simple BASIC interpreters also now exist, including Microsoft's Small Basic. Many versions of BASIC are also now available for smartphones and tablets via the Apple App Store, or Google Play store for Android. On game consoles, an application for the Nintendo 3DS and Nintendo DSi called Petit Computer allows for programming in a slightly modified version of BASIC with DS button support. A version has also been released for Nintendo Switch. Calculators Variants of BASIC are available on graphing and otherwise programmable calculators made by Texas Instruments, HP, Casio, and others. Windows command-line QBasic, a version of Microsoft QuickBASIC without the linker to make EXE files, is present in the Windows NT and DOS-Windows 95 streams of operating systems and can be obtained for more recent releases like Windows 7 which do not have them. Prior to DOS 5, the Basic interpreter was GW-Basic. QuickBasic is part of a series of three languages issued by Microsoft for the home and office power user and small-scale professional development; QuickC and QuickPascal are the other two. For Windows 95 and 98, which do not have QBasic installed by default, they can be copied from the installation disc, which will have a set of directories for old and optional software; other missing commands like Exe2Bin and others are in these same directories. Other BASIC came to some video game systems, such as the Nintendo Famicom. The various Microsoft, Lotus, and Corel office suites and related products are programmable with Visual Basic in one form or another, including LotusScript, which is very similar to VBA 6. The Host Explorer terminal emulator uses WWB as a macro language; or more recently the programme and the suite in which it is contained is programmable in an in-house Basic variant known as Hummingbird Basic. The VBScript variant is used for programming web content, Outlook 97, Internet Explorer, and the Windows Script Host. WSH also has a Visual Basic for Applications (VBA) engine installed as the third of the default engines along with VBScript, JScript, and the numerous proprietary or open source engines which can be installed like PerlScript, a couple of Rexx-based engines, Python, Ruby, Tcl, Delphi, XLNT, PHP, and others; meaning that the two versions of Basic can be used along with the other mentioned languages, as well as LotusScript, in a WSF file, through the component object model, and other WSH and VBA constructions. VBScript is one of the languages that can be accessed by the 4Dos, 4NT, and Take Command enhanced shells. SaxBasic and WWB are also very similar to the Visual Basic line of Basic implementations. The pre-Office 97 macro language for Microsoft Word is known as WordBASIC. Excel 4 and 5 use Visual Basic itself as a macro language. Chipmunk Basic, an old-school interpreter similar to BASICs of the 1970s, is available for Linux, Microsoft Windows and macOS. Legacy The ubiquity of BASIC interpreters on personal computers was such that textbooks once included simple "Try It In BASIC" exercises that encouraged students to experiment with mathematical and computational concepts on classroom or home computers. Popular computer magazines of the day typically included type-in programs. Futurist and sci-fi writer David Brin mourned the loss of ubiquitous BASIC in a 2006 Salon article[30] as have others who first used computers during this era. In turn, the article prompted Microsoft to develop and release Small Basic;[31] it also inspired similar projects like Basic-256.[32] Dartmouth held a 50th anniversary celebration for BASIC on 1 May 2014,[33] as did other organisations; at least one organisation of VBA programmers organised a 35th anniversary observance in 1999.[34] Dartmouth College celebrated the 50th anniversary of the BASIC language with a day of events[35] on April 30, 2014. A short documentary film was produced for the event.[36] Syntax Typical BASIC keywords Data manipulation LET assigns a value (which may be the result of an expression) to a variable. In most dialects of BASIC, LET is optional, and a line with no other identifiable keyword will assume the keyword to be LET. DATA holds a list of values which are assigned sequentially using the READ command. READ reads a value from a DATA statement and assigns it to a variable. An internal pointer keeps track of the last DATA element that was read and moves it one position forward with each READ. Most dialects allow multiple variables as parameters, reading several values in a single operation. RESTORE resets the internal pointer to the first DATA statement, allowing the program to begin READing from the first value. Many dialects allow an optional line number or ordinal value to allow the pointer to be reset to a selected location. DIM Sets up an array. Program flow control IF ... THEN ... {ELSE} used to perform comparisons or make decisions. Early dialects only allowed a line number after the THEN, but later versions allowed any valid statement to follow. ELSE was not widely supported, especially in earlier versions. FOR ... TO ... {STEP} ... NEXT repeat a section of code a given number of times. A variable that acts as a counter, the "index", is available within the loop. WHILE ... WEND and REPEAT ... UNTIL repeat a section of code while the specified condition is true. The condition may be evaluated before each iteration of the loop, or after. Both of these commands are found mostly in later dialects. DO ... LOOP {WHILE} or {UNTIL} repeat a section of code indefinitely or while/until the specified condition is true. The condition may be evaluated before each iteration of the loop, or after. Similar to WHILE, these keywords are mostly found in later dialects. GOTO jumps to a numbered or labelled line in the program. Most dialects also allowed the form GO TO. GOSUB jumps to a numbered or labelled line, executes the code it finds there until it reaches a RETURN command, on which it jumps back to the statement following the GOSUB, either after a colon, or on the next line. This is used to implement subroutines. ON ... GOTO/GOSUB chooses where to jump based on the specified conditions. See Switch statement for other forms. DEF FN a pair of keywords introduced in the early 1960s to define functions. The original BASIC functions were modelled on FORTRAN single-line functions. BASIC functions were one expression with variable arguments, rather than subroutines, with a syntax on the model of DEF FND(x) = x*x at the beginning of a program. Function names were originally restricted to FN, plus one letter, i.e., FNA, FNB ... Input and output LIST displays the full source code of the current program. PRINT displays a message on the screen or other output device. INPUT asks the user to enter the value of a variable. The statement may include a prompt message. TAB used with PRINT to set the position where the next character will be shown on the screen or printed on paper. AT is an alternative form. SPC prints out a number of space characters. Similar in concept to TAB but moves by a number of additional spaces from the current column rather than moving to a specified column. Mathematical functions ABS Absolute value ATN Arctangent (result in radians) COS Cosine (argument in radians) EXP Exponential function INT Integer part (typically floor function) LOG Natural logarithm RND Random number generation SIN Sine (argument in radians) SQR Square root TAN Tangent (argument in radians) Miscellaneous REM holds a programmer's comment or REMark; often used to give a title to the program and to help identify the purpose of a given section of code. USR transfers program control to a machine language subroutine, usually entered as an alphanumeric string or in a list of DATA statements. CALL alternative form of USR found in some dialects. Does not require an artificial parameter to complete the function-like syntax of USR, and has a clearly defined method of calling different routines in memory. TRON / TROFF turns on display of each line number as it is run ("TRace ON"). This was useful for debugging or correcting of problems in a program. TROFF turns it back off again. ASM some compilers such as Freebasic,[37] Purebasic,[38] and Powerbasic[39] also support inline assembly language, allowing the programmer to intermix high-level and low-level code, typically prefixed with "ASM" or "!" statements. Data types and variables Minimal versions of BASIC had only integer variables and one- or two-letter variable names, which minimized requirements of limited and expensive memory (RAM). More powerful versions had floating-point arithmetic, and variables could be labelled with names six or more characters long. There were some problems and restrictions in early implementations; for example, Applesoft BASIC allowed variable names to be several characters long, but only the first two were significant, thus it was possible to inadvertently write a program with variables "LOSS" and "LOAN", which would be treated as being the same; assigning a value to "LOAN" would silently overwrite the value intended as "LOSS". Keywords could not be used in variables in many early BASICs; "SCORE" would be interpreted as "SC" OR "E", where OR was a keyword. String variables are usually distinguished in many microcomputer dialects by having $ suffixed to their name as a sigil, and values are often identified as strings by being delimited by "double quotation marks". Arrays in BASIC could contain integers, floating point or string variables. Some dialects of BASIC supported matrices and matrix operations, which can be used to solve sets of simultaneous linear algebraic equations. These dialects would directly support matrix operations such as assignment, addition, multiplication (of compatible matrix types), and evaluation of a determinant. Many microcomputer BASICs did not support this data type; matrix operations were still possible, but had to be programmed explicitly on array elements. Examples A simple game implemented in BASIC Unstructured BASIC New BASIC programmers on a home computer might start with a simple program, perhaps using the language's PRINT statement to display a message on the screen; a well-known and often-replicated example is Kernighan and Ritchie's "Hello, World!" program: 10 PRINT "Hello, World!" 20 END An infinite loop could be used to fill the display with the message: 10 PRINT "Hello, World!" 20 GOTO 10 Note that the END statement is optional and has no action in most dialects of BASIC. It was not always included, as is the case in this example. This same program can be modified to print a fixed number of messages using the common FOR...NEXT statement: 10 LET N=10 20 FOR I=1 TO N 30 PRINT "Hello, World!" 40 NEXT I Most first-generation BASIC versions, such as MSX BASIC and GW-BASIC, supported simple data types, loop cycles, and arrays. The following example is written for GW-BASIC, but will work in most versions of BASIC with minimal changes: 10 INPUT "What is your name: "; U$ 20 PRINT "Hello "; U$ 30 INPUT "How many stars do you want: "; N 40 S$ = "" 50 FOR I = 1 TO N 60 S$ = S$ + "*" 70 NEXT I 80 PRINT S$ 90 INPUT "Do you want more stars? "; A$ 100 IF LEN(A$) = 0 THEN GOTO 90 110 A$ = LEFT$(A$, 1) 120 IF A$ = "Y" OR A$ = "y" THEN GOTO 30 130 PRINT "Goodbye "; U$ 140 END The resulting dialog might resemble: What is your name: Mike Hello Mike How many stars do you want: 7 ******* Do you want more stars? yes How many stars do you want: 3 *** Do you want more stars? no Goodbye Mike The original Dartmouth Basic was unusual in having a matrix keyword, MAT.[f] Although not implemented by most later microprocessor derivatives, it is used in this example from the 1968 manual[40] which averages the numbers that are input: 5 LET S = 0 10 MAT INPUT V 20 LET N = NUM 30 IF N = 0 THEN 99 40 FOR I = 1 TO N 45 LET S = S + V(I) 50 NEXT I 60 PRINT S/N 70 GO TO 5 99 END Structured BASIC Second-generation BASICs (for example, VAX Basic, SuperBASIC, True BASIC, QuickBASIC, BBC BASIC, Pick BASIC, PowerBASIC, Liberty BASIC and (arguably) COMAL) introduced a number of features into the language, primarily related to structured and procedure-oriented programming. Usually, line numbering is omitted from the language and replaced with labels (for GOTO) and procedures to encourage easier and more flexible design.[41] In addition keywords and structures to support repetition, selection and procedures with local variables were introduced. The following example is in Microsoft QuickBASIC: REM QuickBASIC example REM Forward declaration - allows the main code to call a REM subroutine that is defined later in the source code DECLARE SUB PrintSomeStars (StarCount!) REM Main program follows INPUT "What is your name: ", UserName$ PRINT "Hello "; UserName$ DO INPUT "How many stars do you want: ", NumStars CALL PrintSomeStars(NumStars) DO INPUT "Do you want more stars? ", Answer$ LOOP UNTIL Answer$ "" Answer$ = LEFT$(Answer$, 1) LOOP WHILE UCASE$(Answer$) = "Y" PRINT "Goodbye "; UserName$ END REM subroutine definition SUB PrintSomeStars (StarCount) REM This procedure uses a local variable called Stars$ Stars$ = STRING$(StarCount, "*") PRINT Stars$ END SUB Object-oriented BASIC Third-generation BASIC dialects such as Visual Basic, Xojo, Gambas, StarOffice Basic, BlitzMax and PureBasic introduced features to support object-oriented and event-driven programming paradigm. Most built-in procedures and functions are now represented as methods of standard objects rather than operators. Also, the operating system became increasingly accessible to the BASIC language. The following example is in Visual Basic .NET: Public Module StarsProgram Private Function Ask(prompt As String) As String Console.Write(prompt) Return Console.ReadLine() End Function Public Sub Main() Dim userName = Ask("What is your name: ") Console.WriteLine("Hello {0}", userName) Dim answer As String Do Dim numStars = CInt(Ask("How many stars do you want: ")) Dim stars As New String("*"c, numStars) Console.WriteLine(stars) Do answer = Ask("Do you want more stars? ") Loop Until answer "" Loop While answer.StartsWith("Y", StringComparison.OrdinalIgnoreCase) Console.WriteLine("Goodbye {0}", userName) End Sub End Module Standards ANSI/ISO/IEC Standard for Minimal BASIC: ANSI X3.60-1978 "For minimal BASIC" ISO/IEC 6373:1984 "Data Processing—Programming Languages—Minimal BASIC" ECMA-55 Minimal BASIC (withdrawn, similar to ANSI X3.60-1978) ANSI/ISO/IEC Standard for Full BASIC: ANSI X3.113-1987 "Programming Languages Full BASIC" INCITS/ISO/IEC 10279-1991 (R2005) "Information Technology – Programming Languages – Full BASIC" ANSI/ISO/IEC Addendum Defining Modules: ANSI X3.113 Interpretations-1992 "BASIC Technical Information Bulletin # 1 Interpretations of ANSI 03.113-1987" ISO/IEC 10279:1991/ Amd 1:1994 "Modules and Single Character Input Enhancement" ECMA-116 BASIC (withdrawn, similar to ANSI X3.113-1987) Compilers and interpreters This section is an excerpt from List of compilers § BASIC compilers.[edit] This list is incomplete; you can help by adding missing items. (June 2016) Compiler Author Active Windows Unix-like Other OSs License type AppGameKit The Game Creators Yes Yes Yes No Proprietary B4X Erel Uziel Yes Yes Yes (Android, iOS) Yes (JVM) BlitzMax Blitz Research No (2015) Yes Yes (Linux, OS X) No DarkBASIC The Game Creators No (2016) Yes No No FreeBASIC FreeBASIC Development Team Yes Yes Yes DOS FutureBASIC Brilor Software Yes No OS X Classic Mac OS Proprietary Gambas Benoît Minisini Yes No Yes No GFA BASIC Frank Ostrowski No (2005) Yes No Amiga, Atari ST, DOS Proprietary Mercury RemObjects Yes Yes (Linux, OS X, Android, iOS) Yes (Web Assembly) Proprietary PowerBASIC (formerly Turbo Basic) PowerBASIC, Inc. No (2010) Yes No DOS Proprietary PureBasic Fantaisie Software Yes Yes Yes Proprietary QB64 Galleon Yes Yes Yes QuickBASIC Microsoft No (1990) No No DOS Proprietary True BASIC True BASIC Yes Yes No Proprietary VSI BASIC for OpenVMS VMS Software, Inc. Yes No No OpenVMS Proprietary Xojo (formerly REALbasic) Xojo Inc. (formerly Real Software) Yes Yes Yes Yes Proprietary This section is an excerpt from List of compilers § BASIC interpreters.[edit] This list is incomplete; you can help by adding missing items. (December 2010) Interpreter Author Windows Unix-like Other OSs License type Bywater BASIC (bwBASIC) Ted Campbell No Yes ? Liberty BASIC Shoptalk Systems Yes No No Proprietary GW-BASIC Microsoft No No MS-DOS Proprietary QBasic Microsoft Yes No No Proprietary Chipmunk Basic Ronald H. Nicholson, Jr. Yes Yes Yes Freeware TI Basic Texas Instruments No No TI-99/4A Proprietary TI Extended Basic Texas Instruments No No TI-99/4A Proprietary Rocky Mountain BASIC [Trans Era] Yes No HP 9000 Proprietary Bas Michael Haardt No Yes No Yabasic Marc-Oliver Ihm Yes Yes No SmallBASIC SmallBASIC Yes Yes Android, OS X SuperBASIC Jan Jones No No Sinclair QL Proprietary PC-BASIC Rob Hagemans Yes Yes OS X See also List of BASIC dialects Notes ^ Fortran's DO had a continue for this purpose, but still required the line number to be entered. ^ Tymshare charged about US$10 per hour (equivalent to $70 in 2021) for accessing their systems. ^ Widely regarded as the first "true" mini, the PDP-8's 12-bit memory space allowed 4,096 address of 12-bits each, or 6,144 bytes. ^ Interpreters are ultimately similar to compilers in the tasks they perform, converting source code to machine code, but differ in when they perform it. Compilers convert the entire program at once and output a separate runnable program. Interpreters generally convert only a single line at a time (or even just a portion of it) and then immediately release that code once the line has completed running. This means they require only enough memory to run a single line, and do not require some form of high-performance secondary memory like a hard drive. ^ Microsoft BASIC left 780 bytes free for user program code and variable values on a 4K machine, and that was running a cut-down version lacking string variables and other functionality. ^ From version 3 onwards.

References ^ Kemeny, John G.; Kurtz, Thomas E. (1964). Basic: a manual for BASIC, the elementary algebraic language designed for use with the Dartmouth Time Sharing System (PDF) (1st ed.). Hanover, N.H.: Dartmouth College Computation Center. ^ "High Math at Hanover". Time. February 23, 1959. ^ a b c d e f Time 2014. ^ Rankin, Joy Lisi (2018), A People's History of Computing in the United States, Cambridge, Massachusetts: Harvard University Press, ISBN 9780674970977, p. 23 ^ "BASIC". Jargon File. Retrieved June 13, 2017. ^ "Thomas E. Kurtz – History of Computer Programming Languages". cis-alumni.org. Retrieved June 13, 2017. ^ Alfred, Randy (January 5, 2008). "May 1, 1964: First Basic Program Runs". Wired. Retrieved June 13, 2017. ^ Brooks, David (11 June 2019). "Finally, a historical marker that talks about something important". Concord Monitor. Archived from the original on 11 June 2019. Retrieved 11 August 2019. ^ Bauer, W. F., Computer design from the programmer's viewpoint Archived July 23, 2016, at the Wayback Machine (Eastern Joint Computer Conference, December 1958) One of the first descriptions of computer time-sharing. ^ "IBM VS the World: That's How It Is". Computerworld. December 5, 1973. ^ Bourne, Charles; Hahn, Trudi Bellardo (August 2003). A History of Online Information Services, 1963–1976. p. 387. ISBN 9780262261753. ^ "2000 Timeshare System". ^ "Passing the 10-year mark". MEASURE Magazine. Hewlett Packard. October 1976. ^ Marks, Shirley (December 1971). The JOSS Years: Reflections on an experiment (PDF) (Technical report). Rand. ^ Savetz, Kevin (April 2013). "Dave Ahl and Betsy Ah" (Interview). ^ a b Ahl, David H. (1973). 101 Basic computer games. Morristown, N.J.: Creative Computing Press. OCLC 896774158. ^ Ahl, David H. (May 11, 1981). "Computer Games". InfoWorld. Vol. 3, no. 9. p. 44. ISSN 0199-6649. ^ Dijkstra, Edsger W. (June 18, 1975). "How do we tell truths that might hurt" (PDF). Selected Writings on Computing: A Personal Perspective. Springer-Verlag (published 1982). pp. 129–131. ISBN 978-0387906522. OCLC 693424350. ^ "We have a BASIC". New Mexico Museum of Natural History and Science. Archived from the original on November 30, 2012. Retrieved April 18, 2007. ^ Pittman, Tom. "you had to pay $5 up front to get it…". www.ittybittycomputers.com. Retrieved June 14, 2017. ^ Ahl, David H. (1979). More basic computer games. Morristown: Creative Computing Press. ISBN 978-0894801372. OCLC 839377789. ^ Ahl, David H. (1984). Big computer games. Morris Plains, N.J.: Creative Computing Press. ISBN 978-0916688400. OCLC 872675092. ^ "Osborne 1". oldcomputers.net. Retrieved June 14, 2017. ^ Lien, David (1981). The BASIC Handbook (Second ed.). San Diego, CA: Compusoft Publishing. p. inside cover. ISBN 0-932760-00-7. ^ Pravin, Jain (2011). The Class Of Java. Pearson Education India. ISBN 9788131755440. ^ "GNE: the C programming language". fysh.org. Retrieved June 14, 2017. During the 1980s, C compilers spread widely, and C became an extremely popular language. ^ "25 years of Delphi and no Oracle in sight: Not a Visual Basic killer but hard to kill". ^ "Product Family Life Cycle Guidelines for Visual Basic 6.0". Msdn2.microsoft.com. March 31, 2005. Retrieved June 16, 2009. ^ "Visual Basic support planned for .NET 5.0". March 11, 2020. ^ Brin, David (September 14, 2006). "Why Johnny Can't Code". Salon. Archived from the original on September 18, 2013. Retrieved June 19, 2017. ^ "Small Basic". Microsoft Developer Network. Archived from the original on March 17, 2014. Retrieved June 19, 2017. ^ "BASIC-256 homepage". ^ "Dartmouth plans celebration for 50th anniversary of BASIC computer language". New Hampshire Union Leader. April 28, 2014. Retrieved June 14, 2017. ^ "50th anniversary of BASIC – Google Search". Google. ^ "BASIC at 50 – Event Schedule". Dartmouth College. Retrieved June 14, 2017. ^ "BASIC at 50". Dartmouth College. Retrieved June 14, 2017. ^ "KeyPgAsm". FreeBasic Wiki. Retrieved August 2, 2017. ^ "Inline x86 ASM". Pure Basic. Retrieved August 2, 2017. ^ "Using assembly-language in your code". Power Basic. Archived from the original on August 2, 2017. Retrieved August 2, 2017. ^ Kemeny, John G.; Kurtz, Thomas E. (January 1968). Basic: a manual for BASIC, the elementary algebraic language designed for use with the Dartmouth Time Sharing System (PDF) (4th ed.). Hanover, N.H.: Dartmouth College Computation Center. p. 53. ^ "Differences Between GW-BASIC and QBasic". May 12, 2003. Archived from the original on October 19, 2013. Retrieved June 28, 2008. General references Sammet, Jean E. (1969). Programming languages: history and fundamentals. Englewood Cliffs, N.J.: Prentice-Hall. OCLC 819683527. Kurtz, Thomas E. (1981). "BASIC". In Wexelblat, Richard (ed.). History of programming languages. History of Programming Languages I. New York: ACM. pp. 515–537. doi:10.1145/800025.1198404. ISBN 978-0127450407. Kemeny, John G.; Kurtz, Thomas E. (1985). Back To BASIC: The History, Corruption, and Future of the Language. Addison-Wesley. p. 141. ISBN 9780201134339. OCLC 11399298. Lien, David A. (1986). The Basic Handbook: Encyclopedia of the BASIC Computer Language (3rd ed.). Compusoft Publishing. ISBN 9780932760333. OCLC 12548310. "Fifty Years of BASIC, the Programming Language That Made Computers Personal". Time. April 29, 2014. External links Wikibooks has a book on the topic of: Programming:BASIC BASIC at Curlie "BASIC—Beginners All-purpose Symbolic Instruction Code". The Encyclopedia of Computer Languages. Murdoch University. The Birth of Basic on YouTube Retrieved from " 2"Texas Instruments home computer" redirects here. For the computer known as the Texas Instruments PC, see Texas Instruments Professional Computer. Home computer by Texas Instruments TI-99/4AManufacturerTexas InstrumentsTypeHome computerRelease dateJune 1981 (1981-06)Introductory priceUS$525 (equivalent to $1,560 in 2021)DiscontinuedMarch 1984Units shipped2.8 million[1]MediaROM cartridgecassettefloppy diskOperating systemTI BASICCPUTMS9900 @ 3 MHzMemory16 KB RAM256 bytes scratchpad RAMGraphicsTMS9918APredecessorTI-99/4 TI-99/4ManufacturerTexas InstrumentsRelease dateOctober 1979 (1979-10)Introductory priceUS$1,150 (equivalent to $4,290 in 2021)Discontinued1981Units shipped~20,000CPUTMS9900 @ 3 MHzGraphicsTMS9918SuccessorTI-99/4A The TI-99/4 and TI-99/4A are home computers released by Texas Instruments in 1979 and 1981, respectively.[2] Based on the Texas Instruments TMS9900 microprocessor originally used in minicomputers, the TI-99/4 was the first 16-bit home computer,[3] and the associated video display controller provided color graphics and sprite support that was among the best of its era. The calculator-style keyboard of the TI-99/4 was cited as a weak point, and TI's reliance on ROM cartridges and their practice of limiting developer information to select third parties resulted in a lack of software for the system. The TI-99/4A was released in June 1981 to address some of these issues, featuring a simplified internal design, full-travel keyboard, improved graphics, and a unique expansion system. At half the price of the original model, sales picked up significantly and TI supported the 4A with peripherals, including a speech synthesizer and a "Peripheral Expansion System" box to contain hardware add-ons. TI released developer information and tools, but the insistence on remaining sole publisher continued to starve the platform of software. The 1981 US launch of the TI-99/4A followed the Commodore VIC-20 by several months. Commodore's CEO Jack Tramiel had once been offended by TI's predatory pricing during the mid-1970s, and retaliated with a price war by repeatedly lowering the price of the VIC-20 and forcing TI to do the same. By 1983, the 99/4A was selling for under US$100, at a loss. Even with the increased user base created by the heavy discounts, Texas Instruments suffered a US$330 million loss in the third quarter of 1983[4] and announced the discontinuation of the TI-99/4A in October 1983, with production ended in March 1984. The TI-99/4 was intended to fit in the middle of a planned range of TI-99 computers, none of which were ultimately released but prototypes and documentation have been rediscovered after the TI-99/4A was discontinued. Features The TI-99/4A is a self-contained console with the motherboard, ROM cartridge slot, and full-travel keyboard in the same case. The power supply is external. An RF modulator allows the use of a television as a monitor. Lowercase letters are displayed as small caps, rather than separate glyphs. TI BASIC, an ANSI-compliant BASIC interpreter based on Dartmouth BASIC, is built-in and includes support for graphics, sound, and file system access. Later versions of the 99/4A, identified by (C)1983 TEXAS INSTRUMENTS V2.2 on the title page, prevent the use of unlicensed ROM cartridges from third-party manufacturers such as Atarisoft. Peripherals include a 5¼" floppy disk drive and controller, an RS-232 card with two serial ports and one parallel port, a P-code card for Pascal support, a thermal printer, a 300-baud acoustic coupler, a tape drive using standard audio cassettes as media, and a 32 KB memory expansion card. 16-bit processor Both TI-99/4 models use the 16-bit TMS9900 CPU running at 3 MHz. The TMS9900 is a single-chip implementation of TI's TI-990 minicomputers. Although a full 16-bit processor, only the system ROM and 256 bytes of scratchpad RAM are available on the 16-bit bus.[5] A computer requires not only a CPU but many support chips for input/output, memory and other systems. TI was a major supplier of such systems, but the majority of these were 8-bit designs. In order to build a complete 16-bit system, the company would have to redesign these many support systems in 16-bit form. Instead, for the computer project they decided to use their existing 8-bit devices for the majority of the system. Only a small portion of the system was 16-bit, and used a second 8-bit computer bus for the rest.[6] One of the key features of the TMS9900 from the minicomputer design that spawned it was the inclusion of several sets of processor registers. Registers are used to store information that is being actively worked on by a particular program, as opposed to main memory which stores much more data but is slower to access. In a minicomputer setting, the system was typically running a time sharing or multitasking operating system, or being used for real-time computing, both of which benefit from being able to quickly switch among programs. To do this, the TMS9900 stored several sets of registers in main memory and could switch between the sets of sixteen 16-bit registers by changing the single workspace pointer register, thereby allowing very rapid context switching. Because the performance of registers is critical to overall machine performance, the new design put 256 bytes of random access memory (RAM) on the 16-bit bus to store up to eight sets of registers. This area of RAM was known as the "scratchpad memory". As the processor's instructions are all 16-bit as well, the 8 KB internal system read only memory (ROM) was also on the 16-bit side.[7] Only the program counter, status register, and workspace pointer registers are actually implemented on the chip itself.[7] Included on the 8-bit side of the system is the majority of the RAM and almost all of the support chips, especially the video display controller (VDP). All accesses to the VDP system are executed eight bits at a time.[8] The system's RAM is managed by the VDP, which provides access to the CPU only when the CPU is not using the memory. This means that user programs and data has to be read over two machine cycles, essentially reducing speed by half. According to IEEE Spectrum, this negates the performance advantage of a 16-bit processor.[6] Video display processor Graphics in the 99/4A are generated by a TMS9918A Video Display Processor (VDP), with a variant for PAL territories. The VDP was developed by Texas Instruments and also sold independently, allowing it to be used in other systems. It serves as the video processor for the ColecoVision and SG-1000 consoles, and an earlier model is part of the MSX computer standard. The TMS9918A supports character-based and bitmap display modes as well as hardware sprites. There are 32 single-color sprites total, but only a maximum of 4 can be displayed per scan line. Each sprite is either 8×8 or 16×16 pixels and can be scaled 2x to 16x16 or 32x32. 16 KB of RAM is provided for the Video Display Processor. VDP RAM is the largest block of writeable memory in the unexpanded TI-99/4A architecture, and is used for storing disk I/O buffers and TI BASIC user programs. Access to this memory has to use the VDP as an intermediary. Expansion All TI-99 models have device drivers built into ROMs in the hardware. When a new peripheral is attached, it is immediately available for any software that wants to use it. All device access uses a generic file-based I/O mechanism, allowing new devices to be added without updating software. The system supports four RS-232 ports and two parallel printer ports. The computer supports two cassette drives through a dedicated port, using a custom data format. Composite video and audio are output through another port on NTSC-based machines, and combine through an external RF modulator for use with a television. PAL-based machines output a more complex YUV signal which is also modulated to UHF externally. Two digital joysticks can be connected through a single DE-9 port. It is identical to the Atari joystick port, but with incompatible pins. Aftermarket adapters allow the use of Atari compatible joysticks.[9] TI sold an official 32 KB RAM expansion.[10] The memory is not available to all uses. For example, an Extended Basic program is restricted to using 24 KB with the remaining 8 KB available for machine code routines. The Mini Memory plug-in module contains 4 KB of battery-backed RAM that can be used as a persistent RAM disk or to load a machine-code program.[11] Peripheral Expansion Box Peripheral Expansion Box or PEB The TI-99/4A can be upgraded via expansion cards added to an eight-slot, external chassis containing its own linear power supply and a full-height 5¼" floppy bay.[12] Encased in silver plastic, but made from sheet steel, this is labeled as the Peripheral Expansion System by TI, but usually called the Peripheral Expansion Box or PEB. Each card has an LED that blinks or flickers when being accessed by software. The section of the power supply that powers the card slots is unregulated. Each card has on-board regulators for its own requirements, which reduces power consumption on a partially-loaded PEB, allowing for cards with unusual voltage requirements. The PEB carries an analog sound input on the expansion bus, allowing the Speech Synthesizer's audio to be carried through the console to the monitor. The audio is also carried through the ribbon cable to the PEB, both allowing the relocation of the Speech Synthesizer to the PEB and the possibility of audio cards offering more features than the console's built-in sound. No official cards from TI do this. Speech synthesizer TI-99/4A speech demo using the built-in vocabulary In the late 1970s and early 1980s, TI was a pioneer in speech synthesis because of its Texas Instruments LPC Speech Chips which were used in its Speak & Spell toys. A plug-in speech synthesizer module was available for the TI-99/4 and 4A. Speech synthesizers were offered free with the purchase of a number of cartridges and were used by video games such as Alpiner and Parsec. Alpiner's speech includes male and female voices and can be sarcastic when the player makes a bad move. The synthesizer uses a variant of linear predictive coding and has a small in-built vocabulary. The original intent was to release small cartridges that plugged directly into the synthesizer unit to increase the device's vocabulary. However, the success of software text-to-speech in the Terminal Emulator II cartridge cancelled that plan.[citation needed] History In 1977, groups within Texas Instruments were designing a video game console, a home computer to compete against the TRS-80 and Apple II, and a high-end business personal computer with a hard drive. The first two groups were both working at TI's consumer products division in Lubbock, Texas, and continually competed. According to Wally Rhines, the 99/4's "ultracheap keyboard" (with calculator-style keys), RF modulator, and ROM cartridges came from the console design. Eventually, the two teams were merged and directed towards the home computer market. Meanwhile, the third team was merged into TI's Data Systems Division, which had a line of minicomputer products and various computer terminals; they viewed the all-in-one machine as a threat and the project was eventually killed.[13] Others within the company persuaded the Lubbock group to use TI's TMS9900 CPU. This was in keeping with TI's "one company, one computer architecture" concept, where a single processor model would scale from consoles to its high-end minicomputers. The TMS9900 is a single-chip implementation of TI's 16-bit TI-990 mini design, and is the CPU in low-end models of that platform.[6] Feature-limited single-chip versions of popular minicomputer designs from the 1960s were popular in the mid-1970s and newly designed 16-bit and 32-bit CPUs like the Intel 8088 and Motorola 68000, respectively, quickly rendered these earlier designs obsolete.[citation needed] Many of the TMS9900's quirky features, like processor registers in main memory, came from its minicomputer roots where such concepts were more common.[citation needed] Meanwhile, another home computer product was emerging from TI's European headquarters, where a third party consulting firm was contracted to produce a prototype codenamed "Mojo". This was based on TI's version of the 8-bit Intel 8080 supported by an all-TI chip set. After a series of discussions, Mojo was abandoned and the Consumer Products concept moved forward.[13] 99/4 In 1979, TI was a successful manufacturer of large computers[6] and was the largest semiconductor manufacturer in the world.[14] Its catalog included a huge variety of analog and digital integrated circuits already widely used in the microcomputers, giving it a single-source advantage no other company could meet. It used this position to take over markets, as it did in the mid-1970s introducing its first scientific calculators. These underpriced its former customers like Commodore and drove them out of the calculator business.[15] Observers expected TI would do the same to the microcomputer market if it released a competitive system.[13] The New York Times suggested that the entry of TI and Hewlett-Packard would reshape the entire industry.[16] Through the development period, several companies attempting to enter the home computer market were faced with significant pushback from the Federal Communications Commission (FCC). The FCC had developed new rules for consumer devices that connected directly to televisions in an effort to control ongoing complaints about interference by poorly shielded devices. Televisions of the era generally had only a single antenna input, and thus connecting to them required the internal video signal of the device to be converted to radio frequency using an RF modulator. The resulting signals were often poorly shielded and signal leakage could be picked up in the antennas of nearby televisions.[17] The new rules were extremely difficult to meet. TI continued battling the FCC both in the lab and in Congress, where it had considerable power due to its position within Texas's high-tech industry. It failed to meet the FCC requirements as the release date approached. The company eventually gave up and bundled a modified Zenith Electronics television as a computer monitor, eliminating the need for the RF modulator that generates the interference by connecting directly to the TV's circuitry using a composite video signal. This put the introductory price at US$1,150 (equivalent to $3,782 in 2021).[18] The machine was met with almost universal disdain when it was released. Every review complained about the keyboard, the lack of lower case characters, any sort of expansion, and almost no software. Sales were almost nonexistent. In July 1980, Adam Osborne reported that, despite poor sales, TI had raised the price of a complete system to $1,400, higher than the popular Apple II, which started at $950. Osborne said, "Some dealers, who have offered the complete system (including the monitor) for less than the price of the Apple, have still been unable to sell it".[19] TI sold fewer than 20,000 computers by summer 1981, less than one tenth Apple or Radio Shack's volume; even Atari, Inc., which reportedly lost $10 million on sales of $13 million of computers, had an Atari 8-bit family installed base more than twice as large.[20] By this time it was clear the machine was a failure. David H. Ahl described the computer as "vastly overpriced, particularly considering its strange keyboard, non-standard Basic, and lack of software".[4] The Times called it an "embarrassing failure".[21] 99/4A Late period, cost-reduced version of the TI-99/4A with beige case Two years after the 99/4's debut, TI released the 99/4A. It adds a typewriter-style keyboard—keeping the non-standard layout—and more expansion options. The expansion system extends from the right side of the chassis, with modules that can be daisy-chained. There is a practical limitation to this, because each module increases the width of the system. The price was initially US$525, less than half that of the 99/4.[4] TI continued lowering the price through 1981, first to $449.95, and then to $399.95 in early 1982, in competition with Commodore's $300 VIC-20. This turned into a price war with Commodore. TI responded by cutting the wholesale price of the 99 by $100, while also offering a $100 rebate directly to consumers, lowering the street price to about $200.[4] TI spokesman Bill Cosby joked how easy it was to sell a computer by paying people $100 to buy one.[21][4] By mid-1982, Jerry Pournelle wrote that TI was "practically giving away the TI-99/4A".[22] An industry joke stated that the company was losing money on each computer, but was making up for it in volume.[4][21] Commodore matched the $200 price in December 1982.[4] TI celebrated the 99/4A's market success at the January 1983 Consumer Electronics Show in Las Vegas.[21] Sales peaked at 30,000 a week that month, but on 10 January 1983 Commodore lowered the price of its computers. In February TI responded with a 99/4A retail price of $150. In April, the VIC-20's bundled retail price reached $100 and the 99/4A followed suit. In the spring of 1983, TI attempted to reduce the parts count to maintain a competitive edge by combining multiple chips into a single custom chip, renaming the 4A PCB as a "QI" (Quality Improved) board and began production of plastic beige cases without the former aluminum trim of the back console. In May, it began offering the PEB for free with the purchase of three peripherals. In August the company reduced prices of peripherals by 50% and offered $100 of free software; in September, it reduced software prices by up to 43%.[21][4][23] The president of Spectravideo later said that "TI got suckered by" Jack Tramiel, head of Commodore.[4] TI was forced to sell the 99/4A for about the same price as the VIC-20, even though it was much more expensive to manufacture. Lack of third-party development TI could not make a profit on the TI-99/4A at a price of $99,[24] but hoped that selling many inexpensive computers would increase sales of more profitable software and peripherals. Because such a razor and blades business model requires that such products be its own,[21] TI strictly controlled development for the computer, discouraging hobbyists and third-party developers[25][22] despite third-party development representing what Pournelle described as "a large unpaid R&D department" for computer companies.[26] A Spinnaker Software executive said that the 99/4A had "the worst software in the business", and Ahl noted that unlike other computers, it did not have "Microsoft BASIC, VisiCalc, WordStar, or any popular games".[4] Peripherals cost about twice as much as for other computers.[24][21] TI joysticks, for example, were of poor quality and difficult to find; one reseller reported that its best-selling product was the Atari CX40 joystick adapter cable.[9] TI did not provide an editor, assembler, or hardware technical information when it released the computer. Pournelle stated that "TI's message is loud and clear: 'Drop dead, hobbyists!'",[26] and added that the company "worked very hard at keeping you outside the machine".[22] Citing Money, publisher of Kilobaud Microcomputing Wayne Green reported in August 1980 that TI planned to have only 100 applications available by the end of 1981, stating that "This tiny figure has to put a chill on the whole industry". Green's company, Instant Software, was a prolific publisher of TRS-80 software, but could not find anyone to port software to the TI. He wrote, "We understand the problems with the system and the efforts Texas Instruments made to make translation difficult".[27] Pournelle added, "TI had rightly concluded that the hobbyists and hackers were a tiny part of the market and wrongly concluded that they were therefore unimportant".[22] Rivals were more open with information. Kilobaud Microcomputing reported that a Commodore executive promised the VIC-20 would have "enough additional documentation to enable an experienced programmer/hobbyist to get inside and let his imagination work".[28][25] IBM learned from TI's mistake, Pournelle said. The company released software and hardware technical information when the IBM PC was announced in 1981,[22] stating that "the definition of a personal computer is third-party hardware and software".[29] TI had also learned from its mistake and no longer ignored hobbyists, Pournelle said in 1982.[22][26] The company insisted on being the sole publisher for the system, however, which many developers refused to agree to.[21] After third-party developers' games for the Atari 2600 became very successful, TI at the June 1983 Consumer Electronics Show announced that only cartridges with a TI-licensed lockout chip would work in the 99/4A. The Boston Phoenix predicted that "most [software developers] just won't bother making TI-compatible versions of their programs",[24] and Pournelle wrote that "TI once again tells the hobbyists to drop dead".[26] No official technical documentation from TI was released until the "Editor/Assembler" development suite was released in 1981, and no system schematics were ever released to the public until after TI had discontinued the computer. Discontinuation After TI in mid-1983 unexpectedly announced a $100 million loss in the second calendar quarter—implying a pretax loss from home computers of $200–250 million—its stock dropped by one third in two days. The Times stated in June 1983 that Cosby's $100 refund "joke is no longer funny", and that "future options are slim". The low price affected the 99/4A's reputation; "When they went to $99, people started asking 'What's wrong with it?'", one retail executive said. An L.F. Rothschild sell-side analyst estimated that TI had prepared to manufacture three million computers in 1983, but would only be able to sell two million.[21] Some observers predicted after the second quarter's loss that the 99/4A would not be able to recover; even if the company did not plan to discontinue the computer, the fear that it would become orphaned technology might cause retailers to avoid ordering inventory.[21] Others thought that TI could sell excess inventory and continue producing the computer.[4] After losing $111 million after taxes in the third calendar quarter of 1983, TI announced plans to discontinue the 99/4A, while continuing to sell the TI Professional MS-DOS-compatible computer.[23] (TI stock rose by 25% after the announcement, because the company's other businesses were strong.)[30] With another TI price cut, retailers sold remaining inventory of the former $1,150 computer during Christmas for $49.[4][31] The 90 Child World stores quickly sold over 40,000 computers[32] at a price referred to as "nearly a stocking stuffer" in a Times article.[33] A total of 2.8 million units were shipped before the TI-99/4A was discontinued in March 1984.[1][34] The 99/4A became the first in a series of home computers to be orphaned by their manufacturer over the next few years, along with the Coleco Adam, Mattel Aquarius, Timex Sinclair 1000, and IBM PCjr. Architecture The TMS9900's machine language instructions must be word-aligned, so at least 16-bits are needed for every instruction. At the time, memory was expensive, so the size of this format was a concern. Additionally, programming the 8-bit side of the system from 16-bit code is somewhat complex. To address this, TI built a pseudo-assembly language known as "Graphic Programming Language", or GPL. This is a compact 8-bit language interpreted by the CPU which dynamically translates the GPL instructions into one or more TMS9900 instructions. GPL also includes utility routines that appear as single instructions in GPL code, allowing complex operations to be reduced to small sequences of code. For example, a block of memory can be cleared with a single instruction. All software originally distributed on ROM cartridges were written using GPL, and are sometimes referred to as GROMs.[7] At the time of launch, the system included only a single user-accessible programming language: TI's built-in BASIC interpreter, written in GPL. On the Creative Computing Benchmark, it runs at roughly half the speed of the Apple II.[35] Technical specifications The TI-99/4A running a program written in BASIC CPU Texas Instruments TMS9900 @ 3 MHz, 16-bit, 64-pin DIP Memory 256 bytes scratchpad RAM for the CPU

16 kB Video Display Processor RAM Video TMS9918A VDP, 40 pin DIP. The earlier 99/4 uses the TMS9918. PAL systems use the "9929" versions of each. 32 single-color sprites in defined layers allowing higher-numbered sprites to transparently flow over lower-numbered sprites. Sprites are available at 8×8 pixels or 16×16 pixels, with a "magnify" bit that doubled all sprites' size but not their resolution. A single bit is available in hardware for collision detection, and the console supports automatic movement via an interrupt routine in the ROM. There can only be 4 visible sprites per scan line. 16 fixed colors (15 visible, one color reserved for "transparent" which shows the background color). Transparent is intended for the 9918's genlock which is disabled in the system. Text mode: 40×24 characters (256 6×8 user-definable characters, no sprites, foreground and background color only, not accessible in BASIC) Graphics mode: 32×24 characters (256 8×8 user-definable characters, full 15 color palette + transparent (available in groups of 8 through the character table) and 32 sprites (The only mode available in BASIC. Extended BASIC is required for sprites, and can access only 28 of them.) Bitmap mode: 256×192 pixels (no more than two colors in an eight-pixel row, full 15 color palette + transparent, all 32 sprites available but interrupt-based motion through the ROM routine is not due to the memory layout, not available to BASIC or the original 9918). Multicolor mode: 64×48 pixels (each pixel may be any color, all 32 sprites are available) All of the above comprise 36 layers starting with the video overlay input, then the background color, then two graphics mode layers, then a layer for each of the 32 sprites. A higher layer obscures a lower layer in hardware, unless that higher layer is transparent. Sound TMS9919, later SN94624, identical to the SN76489 used in many other systems 3 voices, 1 noise (white or periodic) Voices generate square waves from 110 Hz to approximately 115 kHz Console ROM includes interrupt-driven music playback Games TI Invaders by Texas Instruments (1981) See also: Category:Texas Instruments TI-99/4A games Roughly 100 games were published for the TI-99/4A, with most published by Texas Instruments.[36] Some of the games released only for the 99/4A are Parsec, Alpiner, Tombstone City: 21st Century, Tunnels of Doom, and The Attack. TI Invaders and Car Wars are TI's renditions of Space Invaders and Head On respectively. Munch Man is Pac-Man, but the title character fills the maze with a pattern rather than emptying it of dots. Tigervision offered a solution to the memory limitation of the standard cartridge slot in the form of a 24kB memory expansion cartridge that attached to the side expansion interface, emulating an expansion device. This allowed the company to implement a larger game completely in machine code, which was used for Espial and Miner 2049er. Exceltec also released two similar side cartridges: Arcturus[37] and Killer Caterpillar. The media criticized the computer's game library as mediocre.[4][36] TI not only discouraged third-party development, including games, but it also failed to license popular arcade games like Zaxxon and Frogger.[21] Unreleased hardware Hex-Bus The Hex-Bus interface was designed in 1982 and intended for commercial release in late 1983. It connects the console to peripherals via a high-speed serial link. Though it is similar to today's USB (plug and play, hot-swappable, etc.), it was never released, with only a small number of prototypes appearing in collector hands after TI pulled out of the market. TI-99/4A successors The TI-99/4 was intended to fit in the middle of a planned range of TI-99 computers, with prototypes and documentation created for other models. Initial plans were for a lower-end TI-99/2 and a more powerful TI-99/8. Later ideas for expanding the range included a bargain-priced TI-99/3, a terminal TI-99/7, and a direct follow-up to the TI-99/4A referred to as either TI-99/4B or TI-99/5.[38] At the time they left the home computer market, TI had been actively developing two successors to the TI-99/4A that went through several prototypes but never entered production. Some of these prototypes are now in the hands of TI-99/4A collectors. Both machines would have been substantially faster than the original TI-99/4A and used the Hex-Bus serial interface. TI-99/2,[39] a 4K RAM, 32K ROM computer with no color, sound, or joystick port and a Mylar keyboard. TI designed the computer in four and a half months to sell for under $100 and compete with the Sinclair ZX81 and Timex Sinclair 1000. Based on the TMS9995 CPU running at 10.7 MHz and with a built-in RF modulator, performance greatly increased when the screen was blank. The University of Southwestern Louisiana developed system software. 99/2 software ran on the 99/4A, but not vice versa. Working prototypes appeared at the January 1983 Consumer Electronic Show (CES).[40] Home-computer prices declined so quickly, however, that by mid-1983 the 99/4A sold for $99.[41][21] The company canceled the 99/2 in April 1983,[23] but planned to exhibit it at the June CES until other companies' press conferences there indicated that competition would increase.[24] TI-99/8 and 99/6.[42][24] The 99/8 reportedly had a $200 wholesale price.[4] Privately shown to dealers but not announced at June CES, it was formally canceled in October 1983. It included 64 kB of RAM[23] expandable to 15 megabytes, a larger keyboard, built-in speech synthesis, built-in UCSD Pascal operating environment, and the full 16-bit data bus available on the expansion port. It was abandoned in the prototype stage. The Multi Emulator Super System is capable of running what are believed[by whom?] to be the system's ROMs. Legacy The Tomy Tutor and its sibling systems are Japanese computers similar in architecture and firmware to the 99/8. Unlike the 99/8, it was released commercially, but sold poorly outside Japan. Portions of the operating system and BASIC code are similar to the 99/8. As of 2020, there is still an annual Chicago TI Faire[43] where people celebrate the TI-99 family of computers. Post-TI development The Myarc Geneve 9640 is an enhanced TI-99/4A clone built by Myarc as a card to fit into the TI Peripheral Expansion System.[44] It uses an IBM PC/XT detached keyboard. Released in 1987, it is similar to the unreleased TI-99/8 system. It includes a 12 MHz TMS9995 processor, enhanced graphics with 80-column text mode, 16-bit wide RAM, MDOS, and is compatible with nearly all TI software and slot-mounted hardware. A toggle switch slows the computer to the speed of the original. The Second Generation CPU card (SGCPU) was released by the System 99 User Group in 1996 as a card to be installed in the PEB.[citation needed] In 2004, a Universal Serial Bus card and Advanced Technology Attachment controller for IDE hard disks for the PEB were released. A range of plug-in cartridge boards have been developed, allowing software projects to be distributed on cartridge.[45][46] The Phoenix G2,[47] was designed in 2010 by Gary Smith, a member of TI-User Group UK. It uses two FPGAs to emulate the entire architecture of the Myarc Geneve 9640 and the TMS9995 microprocessor. It incorporates an SD card reader, Ethernet, VGA output, and 64 MB RAM. An FPGA-based TMS9918 compatible graphics chip, called the F18A, is a drop-in replacement for the original 9918 VDP, but features VGA output, bypassing the TMS9918A's native composite output, and contains other enhancements such as removal of the restriction of 4 sprites per scan line.[48] See also Texas Instruments Compact Computer 40, a small portable computer introduced in 1983 Texas Instruments Professional Computer (TIPC or TI PC), a personal computer that used the DOS operating system but was not fully compatible with the IBM PC (1983 – c. 1985) Texas Instruments Professional Portable Computer, a contemporaneous portable version of the TI Professional Computer References ^ a b Steve's Old Computer Museum! ^ Bryan Roppolo Boulder. "1979 TI-99/4 Home Computer Literature". Ti994.com. Retrieved 2019-10-28. ^ Texas Instruments TI-99/4, First 16-bit Home Computer, Old-Computers.com, retrieved 23 September 2014 ^ a b c d e f g h i j k l m n Ahl, David H. (March 1984). "Texas Instruments". Creative Computing. pp. 30–32. Retrieved 6 February 2015. ^ TI-99/4A Console Technical Data. Texas Instruments Inc. 1983. p. 10. ^ a b c d Rhines, Walden (22 June 2017). "The Inside Story of Texas Instruments' Biggest Blunder: The TMS9900 Microprocessor". IEEE Spectrum. ^ a b c "The TI-99/4A internal architecture". 16 August 2000. ^ TI-99/4A Console Technical Data. Texas Instruments Inc. 1983. p. 4. ^ a b Mace, Scott (1984-04-09). "Atarisoft vs. Commodore". InfoWorld. p. 50. Retrieved 4 February 2015. ^ "Getting Started with the TI-99/4A". 1983. ^ Getting Started with the TI-99/4A, 1983 ^ "TI-99/4A user-dismantled PEB", 99er ^ a b c Rhines, Walden C. (2017-06-22). "The Texas Instruments 99/4: World's First 16-Bit Home Computer". IEEE Spectrum. Retrieved 2017-07-08. ^ "1980s Trends in the Semiconductor Industry". Semiconductor History Museum of Japan. Retrieved 10 July 2019. ^ Pollack, Andrew (14 January 1984). "Founder of Commodore Resigns Unexpectedly". The New York Times. ^ Schuyten, Peter (6 December 1978). "The Computer Entering Home". The New York Times. ^ Federal Communications Commission Reports: Decisions, Reports, and Orders of the Federal Communications Commission of the United States (Extensive coverage of the FCC's dealings with consumer electronics is covered in this later work). FCC. 1983. ^ Knight, Daniel (19 December 2015). "Texas Instruments' Personal Computers". ^ Osborne, Adam (1980-07-07). "Radio Shack's Videotex". InfoWorld. pp. 9, 28. Retrieved 15 February 2016. ^ Hogan, Thom (1981-09-14). "State of Microcomputing / Some Horses Running Neck and Neck". pp. 10–12. Retrieved 2019-04-08. ^ a b c d e f g h i j k l Pollack, Andrew (1983-06-19). "The Coming Crisis in Home Computers". The New York Times. Retrieved 19 January 2015. ^ a b c d e f Pournelle, Jerry (July 1982). "Computers for Humanity". BYTE. p. 392. Retrieved 19 October 2013. ^ a b c d Mace, Scott (1983-11-21). "TI retires from home-computer market". InfoWorld. pp. 22, 27. Retrieved 2011-02-25. ^ a b c d e Mitchell, Peter W. (1983-09-06). "A summer-CES report". Boston Phoenix. p. 4. Retrieved 10 January 2015. ^ a b Thornburg, David D. (April 1981). "The Commodore VIC-20: A First Look". Compute!. p. 26. ^ a b c d Pournelle, Jerry (July 1983). "Interstellar Drives, Osborne Accessories, DEDICATE/32, and Death Valley". BYTE. p. 340. Retrieved 28 August 2016. ^ Green, Wayne (August 1980). "Publisher's Remarks". Kilobaud. p. 8. Retrieved 23 June 2014. ^ "Commodore: New Products, New Philosophies". Kilobaud. September 1980. pp. 26–28. Retrieved 23 June 2014. ^ Bunnell, David (April–May 1982). "Boca Diary". PC Magazine. p. 22. Retrieved 21 October 2013. ^ "IBM's Peanut Begins New Computer Phase". Boston Globe. Associated Press. 1983-11-01. p. 1. ^ Kleinfield, N. R. (1984-12-22). "Trading Up in Computer Gifts". The New York Times. Retrieved 5 February 2015. ^ Rosenberg, Ronald (1983-12-08). "Home Computer? Maybe Next Year". The Boston Globe. ^ "Under 1983 Christmas Tree, Expect the Home Computer". The New York Times. 1983-12-10. ISSN 0362-4331. Retrieved 2017-07-02. ^ TI-99 Home Computer Timeline Bill Gaskill ^ Knight, Daniel (10 January 2016). "How Fast Were Those Late 1970s Home Computers?". Low End Mac. ^ a b Mace, Scott (1984-05-07). "In Praise of Classics". InfoWorld. p. 56. Retrieved 6 February 2015. ^ "Cartridge pictures", TI-99/4A home computer, Hex bus ^ Shaw, Stephen. "Home Computing with Texas Instruments". shawweb.myzen.co.uk. Retrieved 26 February 2022. ^ "99/2", 99er ^ Littlejohn, Harry; Jander, Mark (June 1983). "Texas Instruments' 99/2 Basic Computer". BYTE. p. 128. Retrieved 19 October 2013. ^ Lock, Robert (June 1983). "Editor's Notes". Compute!. p. 6. Retrieved 30 October 2013. ^ "99/8", 99er ^ "Faire", TI-99, Main byte ^ OldComputers (online museum) ^ "Hardware projects", TI-99/4A home computer, Hex bus ^ You Tube ^ "G2", TI-99 UG, UK, archived from the original on 2010-09-19 ^ "Archives", Code hack create External links Media related to Texas Instruments TI-99/4A at Wikimedia Commons Ninerpedia wiki devoted to the TI-99 series 1979 TI-99/4 The TI-99/4A Home Computer Page Portal: 1980s Retrieved from "