Click to prove you're human



```
To learn anything effectively, practicing and solving problems is essential. To help you master C programming, we have compiled over 100 C programming, pointers, and more. These C Examples cover a range of
questions, from fundamental concepts to advanced topics, and are frequently asked in C-based programming interviews. C program Topics: Prerequisite: Before you start with these C programming is a versatile language that powers many foundational
 systems. C Program - Basic Program - Control Flow Program - Control Flow Program - Pointers Program - Structures and Unions Program - Structures and Unions Program - Structures and Unions Program - File IOC Program - Pointers Program - Pointers Program - Structures and Unions Program - Structures and Unions Program - Structures and Unions Program - Pointers Program - Poin
 platform" redirects here. For the railway station interchange, see cross-platform interchange, see cross-platform television, see multi-platform television. Virtualbox about screen with representations of all major desktop operating system platforms Within
computing, cross-platform software (also called multi-platform software, or platform-independent software requires a separate build for each platform, but some can be directly run on any platform without
 special preparation, being written in an interpreted language or compiled to portable bytecode for which the interpreters or run-time packages are common or standard components of all supported platform software may run on many run on many run on tinux, macOS and Microsoft Windows. Cross-platform software may run on many run on many run on tinux, macOS and Microsoft Windows.
 platforms, or as few as two. Some frameworks for cross-platform development are Codename One, ArkUI-X, Kivy, Qt, GTK, Flutter, NativeScript, Xamarin, Apache Cordova, Ionic, and React Native. [3] Main article: Computing platform Platform can refer to the type of processor (CPU) or other hardware on which an operating system (OS) or application
runs, the type of OS, or a combination of the two.[4] An example of a common platform is Android which runs on the ARM architecture family. Other well-known platform are Linux/Unix, macOS and Windows, these are all cross-platform.[4] Applications can be written to depend on the features of a particular platform—either the hardware, OS, or
virtual machine (VM) it runs on. For example, the Java platform is a common VM platform which runs on many OSs and hardware types. A hardware platform can refer to an instruction set architecture. For example: ARM or the x86 architecture. These machines can run different operating systems. Smartphones and tablets generally run ARM
architecture, these often run Android or iOS and other mobile operating systems. A software platform can be either an operating system (OS) or programming environment, though more commonly it is a combination of both. An exception is Java, which uses an OS-independent virtual machine (VM) to execute Java bytecode. Some software platforms
are: Android (ARM64) ChromeOS (ARM32, ARM64, IA-32, x86-64) Common Language Infrastructure (CLI) by Microsoft, implemented in: The legacy .NET framework (simply called ".NET") that works across Microsoft Windows, macOS, and Linux. Other implementations such as
Mono (formerly by Novell and Xamarin[5]) HarmonyOS (ARM64, RISC-V, x86, x64, and LoongArch, iPadOS (ARMv8-A)) iPadOS (ARMv8-A) java Linux (Alpha, ARC, ARM, C-Sky, Hexagon, LoongArch, m68k, Microsoft C to P-Code (1980-1982: many
architectures and systems) macOS x86, ARM (Apple silicon) Microsoft Windows (IA-32, x86-64, ARM, ARM64) PlayStation 4 (x86), PlayStation 4 (x86), PlayStation 3 (PowerPC) and PlayStation 5 (ARM) Solaris (SPARC Unix (many platforms since 1969) Web browsers - mostly compatible with each other, running JavaScript web-apps Xbox Minor, historical
 AmigaOS (m68k), AmigaOS 4 (PowerPC), AROS (x86, PowerPC, m68k), MorphOS (PowerPC, m68k), MorphOS (PowerPC), are language
is typically compiled to run on a VM that is part of the Java platform. The Java virtual machine (Java VM, JVM) is a CPU implement a JVM. Java software can be executed by a hardware-based Java processor. This is used mostly in embedded
systems. Java code running in the JVM has access to OS-related services, like disk input/output (I/O) and network access, if the appropriate privileges are granted. The JVM makes the system calls on behalf of the Java application. This lets users to decide the appropriate privileges are granted. The JVM makes the system calls on behalf of the Java application. This lets users to decide the appropriate privileges are granted.
 network access is usually enabled for desktop applications, but not for browser-based applets. The Java Native Interface (JNI) can also be used to access OS-specific functions, with a loss of portability. Currently, Java Standard Edition software can run on Microsoft Windows, macOS, several Unix-like OSs, and several real-time operating systems for
embedded devices. For mobile applications, browser plugins are used for Windows and Mac based devices, and Android has built-in support for Java. There are also subsets of Java, such as Java Card or Java Platform, Micro Edition, designed for resource-constrained devices. For software to be considered cross-platform, it must function on more than
one computer architecture or OS. Developing such software can be a time-consuming task because different of Ss have different application programming interfaces (API). Software written in a popular programming language such as C or
C++, it does not mean it will run on all OSs that support that language—or even on different versions of the same OS. Web applications are typically described as cross-platform. Web applications generally employ a client-server model, but vary widely in
complexity and functionality. It can be hard to reconcile the desire for features with the need for compatibility. Basic web applications perform all or most processing from a stateless server, and pass the result to the client web browser. All user interaction with the application consists of simple exchanges of data requests and server responses. This
type of application was the norm in the early phases of World Wide Web application development. Such application model, identical to that of serving static web pages. Today, they are deemed more critical than advanced functionality
Prominent examples of advanced web applications include the Web interface to Gmail and Google Maps. Such applications routinely depend on additional features found only in the more recent versions of popular web browsers. These features include Ajax, JavaScript, Dynamic HTML, SVG, and other components of rich web applications. Because of
the competing interests of compatibility and functionality, numerous design strategies have emerged. Many software systems use a layered architecture where platform-dependent code is restricted to the upper- and lowermost layers. Graceful degradation attempts to provide the same or similar functionality to all users and platforms, while
diminishing that functionality to a least common denominator for more limited client browsers. For example, a user attempting to use a limited-feature browser to access Gmail may notice that Gmail switches to basic mode, with reduced functionality but still of use. Some software is maintained in distinct codebases for different (hardware and OS)
 platforms, with equivalent functionality. This requires more effort to maintain the code, but can be worthwhile where the amount of platform-specific formats. One technique is conditional compilation. With this technique, code that is common
to all platforms is not repeated. Blocks of code that are only relevant to certain platforms are made conditional, so that they are only interpreted or compiled when needed. Another technique is separation of functionality, which disables functionality not supported by browsers or OSs, while still delivering a complete application to the user. (See also:
 Separation of concerns.) This technique is used in web development where interpreted code (as in scripting languages) can query the platform it is running on to execute different blocks conditionally.[6] Third-party libraries attempt to simplify cross-platform capability by hiding the complexities of client differentiation behind a single, unified API, at
the expense of vendor lock-in. Responsive web design (RWD) is a Web design approach aimed at crafting the visual layout of sites to provide an optimal viewing experience—easy reading and navigation with a minimum of resizing, panning, and scrolling—across a wide range of devices, from mobile phones to desktop computer monitors. Little or no
platform-specific code is used with this technique. Cross-platform applications need much more integration testing. Some web browsers prohibit installation of different versions on the same machine. There are several approaches used to target multiple platforms, but all of them result in software that requires substantial manual effort for testing and the same machine.
 maintenance.[7] Techniques such as full virtualization are sometimes used as a workaround for this problem. Tools such as the Page Object Model allow cross-platform tests to be scripted so that one test case covers multiple versions are
becoming increasingly popular but many computer users still use traditional applications overlap and blur the distinction between traditional applications overlap and blur the distinction
Nevertheless, this simplifying distinction is a common and useful generalization. Traditional application software has been distributed as binary files, especially executable forms they were built for—which means that a single cross-platform executable forms they were built for—which means that a single cross-platform executable forms they were built for—which means that a single cross-platform executable forms they were built for—which means that a single cross-platform executable forms they were built for—which means that a single cross-platform executable forms they were built for—which means that a single cross-platform executable forms they were built for—which means that a single cross-platform executable forms they were built for—which means that a single cross-platform executable forms they were built for—which means that a single cross-platform executable forms they were built for—which means that a single cross-platform executable forms they were built for—which means that a single cross-platform executable forms they were built for—which means the forms the forms they were built forms the f
particular platform. Instead, generally there is a selection of executable, each built for one platform, using a toolset that translates—transcompiles—a single codebase into multiple binary executables. For
example, Firefox, an open-source web browser, is available on Windows, macOS, Linux, and BSD on multiple computer architectures. The four platforms (in this case, Windows, macOS, Linux, and BSD) are separate executable distributions, although they come largely from the
 same source code. In rare cases, executable code built for several platforms is combined into a single executables for different platforms. In this case, programmers must port the source code to the new platform. For example, an application such as
 Firefox, which already runs on Windows on the x86 family, can be modified and re-built to run on Linux on the x86 (and potentially other architectures) as well. The multiple versions of the code may be stored as separate codebases, or merged into one codebase. An alternative to porting is cross-platform virtualization, where applications compiled for
one platform can run on another without modification of the source code or binaries. As an example, Apple's Rosetta, which is built into Intel-based Macintosh computers, runs applications compiled for the previous generation of Macs that used PowerPC CPUs. Another example is IBM PowerVM Lx86, which allows Linux/x86 applications to run
unmodified on the Linux/Power OS. Example of cross-platform binary software: The LibreOffice office suite is built for Microsoft Windows, macOS, Linux, FreeBSD, OpenBSD, Android, iOS, iPadOS, ChromeOS, web-based Collabora Online and many others.[8][9] Many of these are supported on several hardware platforms with processor
architectures including IA-32, x86-64, ARM (ARMel, ARMhf, ARM64), MIPS, MIPSel, PowerPC, ppc64le, and S390x[9][10] A script can be considered to be cross-platform if its interpreter is available on multiple platforms and the script only uses the facilities built into the language. For example, a script written in Python for a Unix-like system will
likely run with little or no modification on Windows, because Python also runs on Windows; indeed there are many implementations (e.g. IronPython for .NET Framework). The same goes for many of the open-source scripting languages. Unlike binary executable files, the same script can be used on all computers that have software to interpret the
script. This is because the script is generally stored in plain text in a text file. There may be some trivial issues, such as the representation of a new line character. Some popular cross-platform scripting languages are: bash - A Unix shell commonly run on Linux and other modern Unix-like systems, as well as on Windows via the Cygwin POSIX
compatibility layer, Git for Windows, or the Windows, or the Windows, or the Windows, or the Windows, and more. PHP - Mostly used for CGI programming, small system administration tasks, and more. PHP - Mostly used for web applications. Python - A language which focuses on rapid applications. Python - A language which focuses on rapid applications.
object-oriented language which aims to be easy to read. Can also be used on the web through Ruby on Rails. Tcl - A dynamic programming language, suitable for a wide range of uses, including web and desktop applications, networking, administration, testing and many more. Cross-platform or multi-platform is a term that can also apply to video
games released on a range of video game consoles. Examples of cross-platform games include: Miner 2049er, Tomb Raider: Legend, FIFA series, NHL series and Minecraft. Each has been released across a variety of gaming platforms, such as the Wii, PlayStation 3, Xbox 360, personal computers, and mobile devices. Some platforms are harder to
 write for than others, requiring more time to develop the video game to the same standard. To offset this, a video game may be released on a few platforms first, then later on others. Typically, this happens when a new gaming system is released on a few platforms first, then later on others.
games may not be cross-platform because of licensing agreements between developers and video game console. As an example, Disney could create a game with the intention of release on the latest Nintendo and Sony game consoles. Should Disney license the game with Sony first, it may
 be required to release the game solely on Sony's console for a short time or indefinitely. Main articles: Cross-platform play and List of video games that support cross-platforms. Psyonix, Epic Games, Microsoft, and Valve all possess technology that
 allows Xbox 360 and PlayStation 3 gamers to play with PC gamers, leaving the decision of which platform to use to consumers. The first game to allow this level of interactivity between PC and console games (Dreamcast with specially produced keyboard and mouse) was Quake 3.[11][12] Games that feature cross-platform online play include Rocket
 League, Final Fantasy XIV, Street Fighter V, Killer Instinct, Paragon and Fable Fortune, and Minecraft with its Better Together update on Windows 10, VR editions, Pocket Edition and Xbox One. Cross-platform programming is the practice of deliberately writing software to work on more than one platform. There are different ways to write a cross-
 platform application. One approach is to create multiple versions of the same software in different source code files and the Macintosh version another, while a FOSS *nix system might have a third. While this is straightforward, compared to
 developing for only one platform it can cost much more to pay a larger team or release products more slowly. It can also result in more bugs to be tracked and fixed. Another approach is to use software that hides the differences between the platforms. This abstraction layer insulates the application from the platform. Such applications are platform
 agnostic. Applications that run on the JVM are built this way. Some applications mix various methods of cross-platform programming to create the final application. An example is the Firefox web browser, which uses abstraction to build some of the lower-level components, with separate source subtrees for implementing platform-specific features
(like the GUI), and the implementation of more than one scripting language to ease software portability. Firefox implements XUL, CSS and JavaScript for extending the browser, in addition to classic Netscape-style browser plugins. Much of the browser itself is written in XUL, CSS, and JavaScript. There are many tools[13][14] available to help the
process of cross-platform programming: 8th: a development language which utilizes Juce as its GUI layer. It currently supports Android, iOS, Windows, macOS, Linux and Raspberry Pi. Anant Computing: A mobile application platform that works in all Indian languages, including their keyboards, and also supports AppWallet and native performance in
all OSs. AppearIQ: a framework that supports the workflow of app development and deployment in an enterprise environment. Natively development of mobile apps that run on different platforms. Boden: a UI
 framework written in C++. Cairo: a free software library used to provide a vector graphics-based, device-independent API. It is designed to provide primitives for 2-dimensional drawing across a number of different backends. Cairo is written in C and has bindings for many programming languages. Cocos2d: an open-source toolkit and game engine for
developing 2D and simple 3D cross-platform games and applications. Codename One: an open-source Write Once Run Anywhere (WORA) framework for Java and Kotlin developers. Delphi: an IDE which uses a Pascal-based language for development. It supports Android, iOS, Windows, macOS, Linux. Ecere SDK: a GUI and 2D/3D graphics toolkit and
IDE, written in eC and with support for additional languages such as C and Python. It supports Linux, FreeBSD, Windows, Android, macOS and the Web through Emscripten or Binaryen [Wikidata] (WebAssembly). Eclipse: an open-source development environment. Implemented in Java with a configurable architecture which supports many tools for
 software development. Add-ons are available for several languages, including Java and C++. FLTK: an open-source toolkit, but more lightweight because it restricts itself to the GUI. Flutter: A cross-platform UI framework for IOS, Android, Mac, Windows and developed by Google. fpGUI: An open-source widget toolkit that is completely implemented
in Object Pascal. It currently supports Linux, Windows and a bit of Windows CE. GeneXus: A Windows CE. GeneXus: A Windows cetion and supporting C#, COBOL, Java including Android and BlackBerry smart devices, Objective-C for Apple mobile devices,
RPG, Ruby, Visual Basic, and Visual FoxPro. GLBasic: A BASIC dialect and compilers for many platforms and supports numerous platform (Windows, Mac, Linux, Android, iOS and some exotic handhelds). Godot: an SDK which uses Godot Engine. GTK+: An open-source widget toolkit for Unix-like
 systems with X11 and Microsoft Windows. Haxe: An open-source language. Juce: An application framework written in C++, used to write native software on numerous systems (Microsoft Windows, POSIX, macOS), with no change to the code. Kivy: an open-source cross-platform UI framework written in Python. It supports Android, iOS, Linux, OS X
Windows and Raspberry Pi. LEADTOOLS: Cross-platform SDK libraries to integrate recognition, document, medical, imaging, and multimedia technologies into Windows, iOS, macOS, Android, Linux and web applications.[15] LiveCode: a commercial cross-platform rapid application development language inspired by HyperTalk. Lazarus: A
programming environment for the FreePascal Compiler. It supports the creation of self-standing graphical and console applications and runs on Linux, MacOSX, iOS, Android, WinCE, Windows and WEB. Max/MSP: A visual programming language that encapsulates platform-independent code with a platform-specific runtime environment into
 applications for macOS and Windows A cross-platform Android runtime. It allows unmodified Android apps to run natively on iOS and macOS Mendix: a cloud-based low-code application development platform. MonoCross: an open-source model-view-controller design pattern where the model and controller are cross-platform but the view is platform.
specific.[16] Mono: An open-source cross-platform version of Microsoft .NET (a framework for applications and programming languages) MoSync: an open-source platform for building macOS, Windows and Linux applications. OpenGL: a 3D
graphics library. Pixel Game Maker MV: A proprietary 2D game development software for Windows and Linux applications. ReNative: The universal development SDK to build multi-platform projects with React Native.
 Includes latest iOS, tvOS, Android, Android, Android TV, Web, Tizen TV, Tizen Watch, LG webOS, macOS/OSX, Windows, KaiOS, Firefox OS and Firefox TV platforms. Qt: an application framework and widget toolkit for Unix-like systems with X11, Microsoft Windows, macOS, and other systems—available under both proprietary and open-source licenses
Simple and Fast Multimedia Library: A multimedia Library written in C that creates an abstraction over various platforms' graphics, sound, and input APIs. It runs on OSs including Linux, Windows and macOS and is
 aimed at games and multimedia applications. Smartface: a native app development tool to create mobile applications for Android and iOS, using WYSIWYG design editor with JavaScript code editor. Tcl/Tk Titanium Mobile: open source cross-platform framework for Android and iOS, using WYSIWYG design editor with JavaScript code editor. Tcl/Tk Titanium Mobile: open source cross-platform framework for Android and iOS, using WYSIWYG design editor with JavaScript code editor. Tcl/Tk Titanium Mobile: open source cross-platform framework for Android and iOS, using WYSIWYG design editor with JavaScript code editor.
includes a set of libraries (GUI, SQL, etc..), and IDE. It supports Windows, macOS, iOS, Android, WebAssembly and Linux using C#. Unreal: A cross-platform SDK which uses Unity Engine. V-Play is a cross-platform development
SDK based on the popular Qt framework. V-Play apps and games are created within Qt Creator. WaveMaker: A low-code development tool to create responsive web and hybrid mobile (Android & iOS) applications. WinDev: an Integrated Development Environment for Windows, Linux, .Net and Java, and web browers. Optimized for business and
 industrial applications. wxWidgets: an open-source widget toolkit that is also an application framework.[17] It runs on Unix-like systems with X11, Microsoft Windows and macOS. Xojo: a RAD IDE that uses an object-oriented programming language to compile desktop, web and iOS apps. Xojo supports natively compiling to Windows, macOS, iOS and
 Linux, and can also create compiled web apps that are able to be run as standalone servers or through CGI. This section possibly contains original research should be removed. (March 2025) (Learn how and when to remove this
message) There are many challenges when developing cross-platform software: Testing cross-platform development as "write once, debug everywhere", a
take on Sun Microsystems' "write once, run anywhere" marketing slogan. Developers are often restricted to using the lowest common denominator subset of features which are available on all platforms. This may hinder the application's performance or prohibit developers from using the most advanced features of each platform. Different platforms
often have different user interface conventions, which cross-platform applications do not always accommodate. For example, applications developed for macOS and GNOME are supposed to place the most important button on the right-hand side of a window or dialog, whereas Microsoft Windows and KDE have the opposite convention. Though many
of these differences are subtle, a cross-platform application which does not conform to these conventions may even result in data loss, such as in a dialog box confirming whether to save or discard changes. Scripting languages and VM bytecode must be translated
 InstallAnywhere address this need. Cross-platform execution environments may suffer cross-platform security flaws, creating a fertile environment for cross-platform malware. [18] Operating context List of widget toolkits Hardware virtualization Language binding Source-to-source compiler Binary-code compatibility Comparison of user features of
 messaging platforms ^ "Design Guidelines: Glossary", java.sun.com. Archived from the original on 2012-02-13. Retrieved 2020-10-18. ^ Lee P Richardson (2016-02-16). "Xamarin vs Ionic: A Mobile, Cross Platform, Shootout". ^ a b "Platform Definition"
 The Linux Information Project. Retrieved 2014-03-27. ^ "About Mono". mono-project.com. Retrieved 2015-12-17. ^ Corti, Sascha P. (October 2011). "Browser and Feature Detection". MSDN Magazine. Retrieved 2015-12-17. ^ Companion
 Proceedings of the 36th International Conference on Software Engineering. pp. 642-645. doi:10.1145/2591062.2591097. hdl:1853/53588. ISBN 9781450327688. S2CID 1903037. ^ Mehrotra, Pranob (2020-12-01). "Collabora Office suite gets a new layout for Android tablets and Chromebooks". XDA-Developers. Retrieved 2021-01-15. Collabora Office suite gets a new layout for Android tablets and Chromebooks". XDA-Developers. Retrieved 2021-01-15. Collabora Office suite gets a new layout for Android tablets and Chromebooks". XDA-Developers. Retrieved 2021-01-15. Collabora Office suite gets a new layout for Android tablets and Chromebooks". XDA-Developers. Retrieved 2021-01-15. Collabora Office suite gets a new layout for Android tablets and Chromebooks". XDA-Developers. Retrieved 2021-01-15. Collabora Office suite gets a new layout for Android tablets and Chromebooks". XDA-Developers. Retrieved 2021-01-15. Collabora Office suite gets a new layout for Android tablets and Chromebooks (Android tablets) and Chromebooks (Android tab
Cribba. Quake III Arena, Giant Bombcast, February 15, 2013. ^ A Closer Look At The Dreamcast Internet Starter Kit ^ The GUI Toolkit, Framework Page ^ "Platform Independent FAQ". Archived from the original on 2008-08-16. Retrieved 2009-04-25. ^ "Cross-Platform SDK Libraries for Recognition, Document, Medical, Imaging, and Multimedia"
 www.leadtools.com. Retrieved 2021-03-03. ^ "12 benefits of Xamarin Cross-platform app development". HeadWorks. 15 Mar 2019. ^ WxWidgets Description ^ Warren, Tom (2020-01-14). "Microsoft bids farewell to Windows 7 and the millions of PCs that still run it". The Verge. Retrieved 2020-02-06. Retrieved from " The best way to learn C
programming language is by hands-on practice. This C Exercise questions that are designed for both beginners and advanced programmers. It covers all major concepts like arrays, pointers, for-loop, and many more. So, Keep it Up! Solve topic-wise C exercise questions to strengthen your weak
 topics.Also, Once you've covered basic C exercises, the GeeksforGeeks Practice Platform is a great place to take on more advanced C coding problems and improve your understanding. C Programming Exercises with solutions to help you practice online and improve your coding efficiency in the C
 language. You can solve these questions online in GeeksforGeeks IDE.Q1: Write a Program to Print "Hello World!" on the Console.In this problem, you have to write a simple program that prints "Hello World!" on the Console Screen.For Example, Output: Hello World!" on the console screen.For Example, Output: Hello World!" on the Console In this problem, you have to write a simple program to find the Sum of two numbers
entered by the user. In this problem, you have to write a program that adds two numbers A and B is: 7 Click here to view the solution. Q3: Write a Program to find the size of int, float, double, and char. In this problem, you have to write a
 program to print the size of the variable. For Example, Output: Size of int = 4Click here to view the solution. Q4: Write a Program to Swap the values of two variables. In this problem, you have to write a program to Swap the values of two variables. In this problem, you have to write a program to Swap the values of two variables. In this problem, you have to write a program to Swap the values of two variables. In this problem, you have to write a program to Swap the values of two variables. In this problem, you have to write a program to Swap the values of two variables. In this problem, you have to write a program to Swap the values of two variables. In this problem, you have to write a program to Swap the values of two variables. In this problem, you have to write a program to Swap the values of two variables. In this problem, you have to write a program to Swap the values of two variables. In this problem, you have to write a program to Swap the values of two variables.
Compound Interest is: 133.099243Click here to view the solution.Q6: Write a Program to check if the given number is even or odd. In this problem, you have to write a program to check whether the given number is even or odd. In this problem, you have to write a program to check whether the given number is even or odd. In this problem, you have to write a program to check whether the given number is even or odd. For Example, Input: Enter a Number: 20utput: even Click here to view the solution. Q7: Write a Program to find the larges are not of the control of
number among three numbers. In this problem, you have to write a program to take three number is: 50Enter a Number1: 5Enter a Number1: 5Enter a Number2: 50Enter a Number3: 10Output: Largest number is: 50Click here to view the solution. Q8: Write a Program to make a simple
 calculator. In this problem, you have to write a program to make a simple calculator that accepts two operands and an operator (+, -, *, /): +Enter operand 1: 7Enter operand 2: 8Output: 15.0Click here to view the solution. Q9: Write a Program to find the factorial of a company of the calculator that accepts two operands and an operator (+, -, *, /): +Enter operand 2: 8Output: 15.0Click here to view the solution. Q9: Write a Program to find the factorial of a company of the calculator that accepts two operands are calculator.
 given number. In this problem, you have to write a program to calculate the factorial (product of all the natural numbers less than or equal to the given number. 50utput: Factorial of 5 is: 120Click here to view the solution. Q10: Write a Program to Convert Binary to
 Decimal.In this problem, you have to write a program to convert the given binary number entered by the user into an equivalent decimal number. For Example, Input: Enter a binary number: 101010010 utput: Decimal number entered by the user into an equivalent decimal number. For Example, Input: Enter a binary number: 101010010 utput: Decimal number is: 169Click here to view the solution. 211: Write a Program to print the Fibonacci series using recursion. In this problem, you
Sum of Natural Numbers using recursion. In this problem, you have to write a program to calculate the sum of natural number n. For Example, Input: Enter value of n: 10Output: 55 Click here to view the solution. Q13: Write a program to find the maximum and minimum of an Array. In this problem, you have to write a program to
 find the maximum and the minimum element of the array of size N given by the user. For Example, Input: 4, 7, 2, 1, 9 Output: Minimum element is: 1 Maximum element is: 1 Maximum element is: 1 Maximum element is: 1 Output: Minimum element is: 1 Output: 1 Ou
 Reversing an array means changing the order of elements so that the first element becomes the last element becomes the second element elemen
array to the left. In this problem, you have to write a program that takes an array arr[] of size N from the user and rotates the array to the left (counter-clockwise direction) by D steps, where D is a positive integer. For Example, Input: arr[] = {1, 2, 3, 4, 5}, D = 2Output: After rotation: 3 4 5 1 2Click here to view the solution.Q16: Write a Program to
remove duplicates from the Sorted array. In this problem, you have to write a program that takes a sorted array arr[] of size N from the user and removes the duplicate elements from the array. For Example, Input: arr[] = {1, 2, 2, 3, 4, 4, 4, 5, 5} Output: array after removal of duplicates: 1 2 3 4 5 Click here to view the solution. Q17: Write a Program to
 search elements in an array (using Binary Search). In this problem, you have to write a program that takes an array arr[] of size N and a target value is found print its index else print 'element is not present in array'. For Example, Input: arr[] = { 2, 3, 4, 10, 40
} target: 10Output: Element is present at index 3Click here to view the solution.Q18: Write a Program to reverse a linked list. For Example, Input: 1->2->3->4->NULLOutput: Reversed
 Linked list: 4->3->2->1->NULLClick here to view the solution.Q18: Write a Program to create a dynamic array on eby one by the user. Print the array elements. For Example, Output: Elements of the array are: 1, 2, 3, 4, 5, Click
 here to view the solution.Q19: Write a Program to find the Transpose of a Matrix. In this problem, you have to write a program to find the transpose of a matrix is formed by interchanging its rows with columns. For Example, Input: matrix A: { {1, 1}
1, 1}, {2, 2, 2}, {3, 3, 3}, {4, 4, 4, 4} }Output: Result matrix is: 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1
 Example, Input: str1 = "hello", str2 = "world"Output: helloworldClick here to view the solution. Q21: Write a Program to check if the given string is a palindrome string or not. In this problem, you have to write a program to check if the given string is a palindrome string is palindrome or not. If the str is palindrome print 'str is a
palindrome' else print 'str is not a palindrome'. A string is said to be palindromee'. A string is the same as the string is a palindrome. A string is said to be palindrome if the reverse of the string is the same as the string is a palindrome.
string str entered by the user and print the first letter of each word in a string. For Example, Input: Enter str: Geeks Output: G f GClick here to view the solution. Q23: Write a program to reverse a string using recursionIn this problem, you have to write a program to reverse a string str entered by the user, and reverse that string means changing
the order of characters in the string so that the last character becomes the first cha
read the number of rows (n) entered by the user and print the half-pyramid pattern of numbers. Half pyramid pattern of numbers having a hypotenuse on the right side. For Example, Input: Enter number of rows: 50 tput: 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 Click here to view the solution. Q25: Write a program to print Pascal's
triangle pattern. In this problem, you have to write a simple program to read the number of rows (n) entered by the user and print Pascal's triangle pattern. Pascal's triangle pattern. Pascal's triangle pattern in which the first row has a single number 1 all rows begin and end with the number of rows (n) entered by the user and print Pascal's triangle pattern.
 above them in the previous row. Pascal's TriangleFor Example, Input: Enter number of rows: 50utput: 1 1 1 2 1 1 3 3 1 1 4 6 4 1Click here to view the solution. Q26: Write a program that takes an array arr[] of size N from the user and sorts the array elements in
 ascending or descending order using insertion sort. For Example, Input: arr[] = \{12, 11, 13, 5, 6\}, N = 5Output: 5 6 11 12 13 Click here to view the solution. Q27: Write a program to sort an array using Quick Sort. In this problem, you have to write a program that takes an array arr[] of size N from the user and sorts the array elements in ascending
order using quick sort. For Example, Input: arr[] = { 19, 17, 15, 12, 16, 18, 4, 11, 13 }, N = 9Output: 4 11 12 13 15 16 17 18 19 Click here to view the solution. Q28: Write a program to sort an array of strings in which all characters are of the same case entered by the user and
sort them alphabetically. For Example, Input: arr[] = Output: After sorting: clanguage, geeksforgeeks, geeksforgeeks, geeksforgeeks of one file to another file. In this problem, you have to write a program that takes user input to enter the filenames for reading and writing. Read the contents of one file to another file. In this problem, you have to write a program that takes user input to enter the filenames for reading and writing. Read the contents of one file to another file. In this problem, you have to write a program to copy the contents of one file to another file. In this problem, you have to write a program to copy the contents of one file to another file. In this problem, you have to write a program to copy the contents of one file to another file. In this problem, you have to write a program to copy the contents of one file to another file. In this problem, you have to write a program to copy the contents of one file to another file. In this problem, you have to write a program to copy the contents of one file to another file. In this problem, you have to write a program to copy the contents of one file to another file. In this problem, you have to write a program to copy the contents of one file to another file. In this problem, you have to write a program to copy the contents of one file to another file. In this problem, you have to write a program to copy the contents of one file to another file. In this problem, you have to write a program to copy the contents of one file to another file. In this problem, you have to write a program to copy the contents of the contents of
file and copy the content to another file. If the file specified for reading does not exist or cannot be opened, display an error message "Cannot open file: file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" and terminate the program else print "Content copied to file name" an
the solution.Q30: Write a program to store information on students using structure. In this problem, you have to write a program that stores information about students using structure. The program that stores information about students using structure.
Numbers, Ages, and Total Marks. Print the information for each student. For Example, Output: Student Records: Name = Student1Roll Number = 1Age = 12Total Marks. Print the information for each student. For Example, Output: Student Records: Name = Student1Roll Number = 5Age = 10Total Marks. Print the information for each student. For Example, Output: Student Records: Name = Student1Roll Number = 1Age = 12Total Marks. Print the information for each student. For Example, Output: Student Records: Name = Student1Roll Number = 1Age = 12Total Marks. Print the information for each student. For Example, Output: Student Records: Name = Student1Roll Number = 1Age = 12Total Marks. Print the information for each student. For Example, Output: Student Records: Name = Student1Roll Number = 1Age = 12Total Marks. Print the information for each student. For Example, Output: Student Records: Name = Stu
understanding of C concepts. Learning C language is made easier with this exercise sheet as it helps you practice all major C concepts. Solving these C exercise questions will take you a step closer to becoming a C programmer. Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix,
transform, and build upon the material for any purpose, even commercially. The license terms. Attribution — You must give appropriate credit, provide a link to the license terms. Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests
the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not
have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation. No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you
 use the material. At Bell Labs, Dennis Ritchie developed the C programming language between 1971 and 1973. C is a mid-level structured-oriented programming languages. There are many applications in which C programming language is used, including
language compilers, operating systems, assemblers, network drivers, text editors, print spoolers, modern applications, language interpreters, databases, and utilities. In this article, you will get the frequently and most asked C programming interview questions for
freshers. C Programming Interview Questions - For Freshers1. Why is C called a mid-level language, i.e. a low-level language, i.e. a low-level language, i.e. a low-level language, and a higher-level language. Using asm or __asm_
 function, can write low-level program and also, we can write high-level programming language. Features of C programming language. Features of C programming language. What are basic programming language.
data types supported in the C Programming Language? In the C programming language, the basic data types are the fundamental building blocks used to declare variable. Here are the basic data types supported in C:For more information, refer to
the article -Data Types in C4. What are tokens in C?Tokens are identifiers or the smallest single unit in a program that is meaningful to the compiler. In C we have the following tokens: Keywords: Predefined or reserved words in the C programming language.
name an identifier in C.Constants: Constants are normal variables that cannot be modified in the program once they are defined. Constants refer to a fixed value. They are also referred to as literals. Strings: Strings in C are an array of character indicates that end with a null character indicates the end of the string; Special Symbols: Some
special symbols in C have some special meaning and thus, they cannot be used for any other purpose in the program. # = {} (), *; [] are the special symbols in C programming language. Operators: Symbols that trigger an action when they are applied to any variable or any other object. Unary, Binary, and ternary operators are used in the C
 Programming language. For more information, refer to the article -Tokens in C5. What do you mean by the scope of the variable? Scope in a programming language is the block or a region where a defined variable has its defined scope. In simple scope in a programming language is the block or a region, the variable is automatically destroyed.
 terms, the scope of a variable is equal to its life in the program. The variable can be declared in three places These are:Local Variables: Inside a given function parameters only. For more information, refer to the article -Scope in C6. What are
 preprocessor directives in C?In C preprocessor directives are considered the built-in predefined functions or macros that act as a directive to the compiler and are executed before the program execution. There are multiple steps involved in writing and executing a program in C. Main types of Preprocessor Directives are Macros, File Inclusion,
 Conditional Compilation, and Other directives like #undef, #pragma, etc. Processor in CFor more information, refer to the article - Preprocessor Directives in C 7. What is the use of static variables in C?Static variables in the C programming language are used to preserve the data values between function calls even after they are out of their scope
Static variables preserve their values in their scope and they can be used again in the program without initialization. C // C program to print ("Initial value of static variable #include int main() { static variable main
value of variable without static %d", x); return 0; } OutputInitial value of static variable of Initial value of variable without static 0For more information, refer to the article - Static Variables in C8. What is the difference between malloc() and malloc() in the C programming language?calloc() and malloc() library functions are used to allocate dynamic
 memory. Dynamic memory is the memory is the memory that is allocated during the runtime of the program from the header file that is used to facilitate dynamic memory allocation in the C Programming language. Parameter Malloc() Calloc() Definition It is a function that creates one block of memory of a fixed size. It is a function that
 assigns more than one block of memory to a single variable. Number of arguments. It takes two arguments. Speedmalloc() function is faster than calloc(). Efficiency. It has low time efficiency. It ha
memory allocation. For more information, refer to the article - Dynamic Memory Allocation in C using malloc(), calloc(), free(), and realloc()9. What do you mean by dangling pointers and how are dangling pointers different from memory leaks in C programming? Pointers and how are dangling pointers and how 
pointers i.e, whenever a pointer is pointing to a memory location and In case the variable is deleted and the pointer variable. In C programming memory leak occurs when we allocate memory with the help of the malloc() or calloc() library function, but we forget to free the
 allocated memory with the help of the free() library function. Memory leak causes the program to use an undefined amount of memory from the RAM which makes it unavailable for other running programs this causes our program to crash.10. Write a program to convert a number to a string with the help of sprintf() function in the C library. C // C
program to convert number to // string using sprintf() #include #include // Driver code int main() { char res[20]; float a = 32.23; sprintf(res, "%f", a); printf("The string for the num is 32.230000For more information, refer to the article - sprintf() in C11. What is recursion in C?Recursion is the
 process of making the function call itself directly or indirectly. A recursive function solves a particular problems. Recursion helps to reduce the length of code and make it more understandable. The recursive function uses a LIFO (Last In First Out) structure
 like a stack. Every recursive call in the program requires extra space in the stack memory. For more information, refer to the article - Recursion12. What is the difference between the local and global variables are declared inside a block or function but global variables are declared outside the block or function to be accessed
 globally.Local VariablesGlobal VariablesGlobal Variables store a garbage value. By default value of the global variables that are declared outside the block or a function. Yariables that are declared outside the block or a function. Yariables that are declared outside the block or a function. Yariables that are declared outside the block or a function. Yariables is destroyed after the block or a function. Yariables that are declared outside the program is
 executed. Variables are stored inside the stack unless they are specified by the programmer. The storage location of the global variable is decided by the compiler. To access the local variables in other functions parameter passing is required. No parameter passing is required. They are globally visible throughout the program. 13. What are pointers and
 their uses? Pointers are used to store the address of the variable or a memory location. Pointer can also be used to refer to another pointers are:To pass arguments by referenceFor accessing array elementsTo return multiple valuesDynamic
                  locationTo implement data structuresTo do system-level programming where memory addresses are usefulWorking of PointerFor more information, refer to the article - Pointer Uses in C.14. What is typedef in C?In C programming, typedef is a keyword that defines an alias for an existing type. Whether it is an integer variable, function
 parameter, or structure declaration, typedef will shorten the name. Syntax:typedef Here, existing type is already given a name is the new name for the existing variable. Example:typedef long long ll15. What are loops and how can we create an infinite loop in C?Loops are used to execute a block of statements repeatedly. The statement
 which is to be repeated will be executed n times inside the loop until the given condition is reached. There are two types of loops Entry controlled and Exit-controlled loops in the C programming language. An Infinite loop is a piece of code that lacks a functional exit. So, it repeats indefinitely. There can be only two things when there is an infinite loop
in the program. One it was designed to loop endlessly until the condition is met within the loop. Another can be wrong or unsatisfied break conditions in the program for infinite loop // using for, while, do-while #include // Driver code int main() { for (;;) { printf("Infinite-loop");
} while (1) { printf("Infinite-loop"); } do { printf("Infinite-loop"); } while (1); return 0; } For more information, refer to the article - Loops in C16. What is the difference between type casting operator. The data
type is converted to another data type, a caste operator is neededIn type conversion, there is no need for a casting in order to cast the data type, a caste operator is neededIn type conversion, there is no need for a casting
 operator. Type casting is more efficient and reliable. Type conversion is less efficient and less reliable than type casting takes place during the program design by the programmer. Type converted; Syntax: int a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; b = a; // a = 20; float b; a = 20; float b;
 20.0000 For more information, refer to the article - Type Casting and Type ConversionC language has numerous libraries which contain predefined functions. All header files must have a '.h' extension. Header files contain function definitions, data type
          itions, and macros which can be imported with the help of the preprocessor directive '#include'. Preprocessor directives instruct the compiler that these files and Pre-existing header files. For example, if our code needs to take
input from the user and print desired output to the screen then 'stdio.h' header file must be included in the program as #include. This header file contains functions and their
types?The function is a block of code that is used to perform a task multiple times in our program. Functions avoid repetition of code and increase the readability of the program. Functions avoid repetition of code and increase the readability of the program.
 functions: User-defined Functions: Functions: Functions that are defined by the user to reduce the complexity of big programs. They are built only to satisfy the condition in which the user is facing issues and are commonly known as "tailor-made functions". Built-in Functions: Library functions are provided by the compiler package and consist of special functions.
 with special and different meanings. These functions give programmers an edge as we can directly use them without defining them. For more information, refer to the article - Functions in C19. What is the difference between macro and functions? A macro is a name that is given to a block of C statements as a pre-processor directive. Macro is defined
with the pre-processed which means that all the macros would be preprocessed but compiled. MacroFunctions are pre-processed but compiled.
using function. Execution speed using a macro is faster. Execution speed using function is slower. The macro doesn't check any Compile-Time Errors. Function check Compile-time errors. For more information, refer to the article
Macro vs Functions 20. How to convert a string to numbers in C?In C we have 2 main methods to convert strings to numbers i.e, Using string stream, Using string stream, Using string to numbers in C?In C we have 2 main methods to convert a string to numbers in c. It reads input from a string rather than standard input string string stream, Using string stream, Using
an argument and an integer value is returned. For more information, refer to the article -String to Numbers in C21. What are reserved keywords? Every keyword is meant to perform a specific task in a program. Their meaning is already defined and cannot be used for purposes other than what they are originally intended for. C Programming language
supports 32 keywords. Some examples of reserved keywords are auto, else, if, long, int, switch, typedef, etc. For more information, refer to the article - Variables and Keywords in C22. What is a structure? The structure is a keyword that is used to create user-defined data types. The structure allows storing multiple types of data in a single unit. The
structure members can only be accessed through the structure variable. Example:struct student { char name[20]; int roll no; char address[20]; char branch[20];} int roll no; char address[20]; int roll no;
char address[50]; char branch[50]; }; // Driver code int main() { struct student obj; strcpy(obj.name, "Kamlesh Joshi"); obj.roll no = 27; strcpy(obj.name); printf("Roll No: %d", obj.roll no); printf("Address: %s", obj.name); printf("Name: %s", obj.name); printf("Roll No: %d", obj.roll no); printf("Address: %s", obj.name); printf("Name: %s", obj.name); printf("Nam
obj.address); printf("Branch: %s", obj.branch); return 0; } OutputName: Kamlesh Joshi Roll No: 27 Address: Haldwani Branch: Computer Science And EngineeringFor more information, refer to the article - Structure in C23. What is union? A union is a user-defined data type that allows users to store multiple types of data in a single unit. However, a
union does not occupy the sum of the memory of all members. It holds the memory of the largest member only. Since the union allocates one common space for all the memory of the largest members. Syntax: union
name of union data type name; \text{}; For more information, refer to the article - Union in C24. What is an r-value and value? An "l-value" will appear either on the right or left side of the assignment operator (=). An "r-value" is a data value
stored in memory at a given address. An "r-value" refers to an object without an identifiable location in memory (i.e. without an address). An "r-value" is an expression that cannot be assigned a value, therefore it can only exist on the right side of an assignment operator (=). Example: int val = 20; Here, val is the 'l-value', and 20 is the 'r-value'. For more
information, refer to the article - r-value and l-value in C25. What is the difference between call by reference between c
function call. Dummy variables copy the address of actual variables. Changes made to dummy variables in the calling function. We can manipulate the actual variables must be stored in
pointer variables. For more information, refer to the article - Call by Value and Call by Reference 26. What is the sleep() function will sleep the present executable for the given amount of time by the thread but other operations of the CPU will
 function properly. sleep() function returns 0 if the requested time has elapsed. For more information, refer to the article - sleep() Function in C27. What are enumerations? In C, enumerations (or enums) are user-defined data types. Enumerations allow integral constants to be named, which makes a program easier to read and maintain. For example,
the days of the week can be defined as an enumeration and can be used anywhere in the program to demonstrate working // of enum in C #include enum week { Mon, Tue, Wed, Thur, Fri, Sat, Sun }; int main() { enum week day; day = Wed; printf("%d", day); return 0; } In
the above example, we declared "day" as the variable, and the value of "Wed" is allocated to day, which is 2. So as a result, 2 is printed. For more information, refer to the article - Enumeration (or enum) in C28: What is a volatile keyword? Volatile keyword is used to prevent the compiler from optimization because their values can't be changed by code
that is outside the scope of current code at any time. The System always reads the current value of a volatile object from the memory location rather than keeping its value in a temporary register at the point it is requested, even if previous instruction is asked for the value from the same object. 29. Write a C program to print the Fibonacci series
using recursion and without using recursion. Fibonacci Numbers C // C program to print Fibonacci (int num, int first, int second, int third) { if (num > 0) { third = first + second; first = second; second = third; printf("%d", third); //
Recursive call for it's // n-1 value Fibonacci(num - 1, first, second, third) } } // Driver code int main() { int num; printf("Fibonacci Series with the help of Recursion:"); printf("Wd %d ", 0, 1); // we are passing n-2 because we have // already printed 2 numbers i.e, 0 and 1 Fibonacci(num - 1, first, second, third) } // Driver code int main() { int num; printf("Wd %d ", 0, 1); // we are passing n-2 because we have // already printed 2 numbers i.e, 0 and 1 Fibonacci(num - 1, first, second, third) }
2, 0, 1, 0); printf("Fibonacci Series without Using Recursion:"); int first = 0, second = 1, third = 0; printf("%d %d ", 0, 1); // Loop will start from 2 because we have // already printed 0 and 1 for (int i = 2; i < num; i++) { third = first + second; printf("%d %d ", 0, 1); // Loop will start from 2 because we have // already printed 0 and 1 for (int i = 2; i < num; i++) { third = first + second; printf("%d %d ", 0, 1); // Loop will start from 2 because we have // already printed 0 and 1 for (int i = 2; i < num; i++) { third = first + second; printf("%d %d ", 0, 1); // Loop will start from 2 because we have // already printed 0 and 1 for (int i = 2; i < num; i++) { third = first + second; printf("%d %d ", 0, 1); // Loop will start from 2 because we have // already printed 0 and 1 for (int i = 2; i < num; i++) { third = first + second; printf("%d %d ", 0, 1); // Loop will start from 2 because we have // already printed 0 and 1 for (int i = 2; i < num; i++) { third = first + second; printf("%d %d ", 0, 1); // Loop will start from 2 because we have // already printed 0 and 1 for (int i = 2; i < num; i++) { third = first + second; printf("%d %d ", 0, 1); // Loop will start from 2 because we have // already printed 0 and 1 for (int i = 2; i < num; i++) { third = first + second; printf("%d %d ", 0, 1); // Loop will start from 2 because we have // already printed 0 and 1 for (int i = 2; i < num; i++) { third = first + second; printf("%d %d ", 0, 1); // Loop will start from 2 because we have // already printed 0 and 1 for (int i = 2; i < num; i++) { third = first + second; printf("%d %d ", 0, 1); // Loop will start from 2 because we have // already printed 0 and 1 for (int i = 2; i < num; i++) { third = first + second; printf("%d %d ", 0, 1); // Loop will start from 2 because we have // already printed 0 and 1 for (int i = 2; i < num; i++) { third = first + second; printf("%d %d ", 0, 1); // Loop will start from 2 because we have // already printed 0 and 1 for (int i = 2; i < num; i++) { third = first + second; print
5Fibonacci Series with the help of Recursion:0 1 1 2 3 Fibonacci Series without Using Recursion:0 1 1 2 3 For more information, refer to the article - Fibonacci Number son. Write a C program to check whether a number is prime #include #i
int num; int check = 1; printf("Enter a number: "); scanf("%d", &num); // Iterating from 2 to sqrt(num) for (int i = 2; i 0) { int rem = n % 10; sum += pow(rem, D); n = n / 10; } // If the sum equals the original number ", var); } else { printf("%d is not an Armstrong number if (var == sum) { printf("%d is an Armstrong number ", var); } // If the sum equals the original number | ("%d is an Armstrong number ", var); } else { printf("%d is an Armstrong number ", var); } // If the sum equals the original number | ("%d is an Armstrong number ", var); } // If the sum equals the original number | ("%d is an Armstrong number ", var); } // If the sum equals the original number | ("%d is an Armstrong number ", var); } // If the sum equals the original number | ("%d is an Armstrong number ", var); } // If the sum equals the original number | ("%d is an Armstrong number ", var); } // If the sum equals the original number | ("%d is an Armstrong number ", var); } // If the sum equals the original number | ("%d is an Armstrong number ", var); } // If the sum equals the original number | ("%d is an Armstrong number ", var); } // If the sum equals the original number | ("%d is an Armstrong number ", var); } // If the sum equals the original number | ("%d is an Armstrong number ", var); } // If the sum equals the original number | ("%d is an Armstrong number ", var); } // If the sum equals | ("%d is an Armstrong number ", var); } // If the sum equals | ("%d is an Armstrong number ", var); } // If the sum equals | ("%d is an Armstrong number ", var); } // If the sum equals | ("%d is an Armstrong number ", var); } // If the sum equals | ("%d is an Armstrong number ", var); } // If the sum equals | ("%d is an Armstrong number ", var); } // If the sum equals | ("%d is an Armstrong number ", var); } // If the sum equals | ("%d is an Armstrong number ", var); } // If the sum equals | ("%d is an Armstrong number ", var); } // If the sum equals | ("%d is an Armstrong number ", var); } // If the sum equals | ("%d is an Armstrong number 
number ", var); } return 0; } OutputEnter number: 82088208 is an Armstrong number43. Write a program to reverse digits // of a number to be reversed : "); scanf("%d", &n); // r will store the remainder while we // reverse the digit and
store it in rev int r = 0; while (n!=0) { r = n % 10; rev = rev * 10 + r; n /= 10; } printf("Number After reversing digits is: 32144. What is the use of an extern storage specifier? The extern keyword in C extends the visibility of variables and functions across
multiple files. It's particularly useful when a variable or function. By default, functions are visible throughout the program, so there is no need to
declare or define extern functions unless they are being used across different files. 45. What is the use of printf() function is used to print the value which is passed as the parameter to it on the console screen. Syntax: printf() functions in C Programming language? Also, explain format specifiers. printf() functions in C Programming language? Also, explain format specifiers. printf() functions in C Programming language? Also, explain format specifiers. printf() functions in C Programming language? Also, explain format specifiers. printf() functions in C Programming language? Also, explain format specifiers. printf() functions in C Programming language? Also, explain format specifiers. printf() functions in C Programming language? Also, explain format specifiers. printf() functions in C Programming language? Also, explain format specifiers. Printf() functions in C Programming language? Also, explain format specifiers. Printf() functions in C Programming language? Also, explain format specifiers. Printf() functions in C Programming language? Also, explain format specifiers. Printf() functions in C Programming language? Also, explain format specifiers. Printf() functions in C Programming language? Also, explain format specifiers. Printf() functions in C Programming language? Also, explain format specifiers. Printf() functions in C Programming language? Also, explain functions in C Programming language? Also, e
reads the values from the console as per the data type of data will be present in the variable during input using scanf() or output using print().%c: Character format specifier used to display and scan character.%d, %i: Signed Integer format specifiers are used to tell the compiler what type of data will be present in the variable during input using scanf() or output using print().%c: Character format specifier used to display and scan character.%d, %i: Signed Integer format specifiers are used to tell the compiler what type of data will be present in the variable during input using scanf() or output using print().%c: Character format specifier used to display and scan character.%d, %i: Signed Integer format specifiers are used to tell the compiler what type of data will be present in the variable during input using scanf() or output using print().%c: Character format specifier used to display and scan character.%d, %i: Signed Integer format specifier used to display and scan character.%d, %i: Signed Integer format specifier used to display and scan character.%d, %i: Signed Integer format specifier used to display and scan character.%d, %i: Signed Integer format specifier used to display and scan character.%d, %i: Signed Integer format specifier used to display and scan character.%d, %i: Signed Integer format specifier used to display and scan character.
specifier used to print or scan an integer value. %f, %e, or %E: Floating-point format specifier is used for Address Printing. For more information, refer to the article - Format Specifier in C46. What is near, far, and huge pointers
in C?Near Pointers: Near pointers are used to store 16-bit addresses only. Using the near pointer of 32 bits size. However, information outside the computer's memory from the current segment can also be accessed. Huge Pointers: Huge pointer is
typically considered a pointer of 32 bits size. But bits located outside or stored outside the segments can also be accessed.47. Mention file operations in C C // C programming Basic File Handling Techniques provide the basic functionalities that programmers can perform against the system. File Operations in C C // C program to check if the linked lister of the linked list of the linked lis
is circular #include #include #include struct Node* next; }; int is Circular(struct Node* next; }; int is Circular (struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular (struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circular if (head == NULL) { return 1; } struct Node* next; // Traversing linked list is null then it is circula
circular else 0 return (ptr == head); } struct Node* newnode(10); head->next = newnode(10); head->next = newnode(12); head->next = newnode(14); head->next->next = newnode(14); head->next->next = newnode(16);
head->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->next->
#include #in
Node* SortedMerge(struct Node* a, struct Node* tail = &dummy; /* a dummy first node to hang the result on */ struct Node* tail = &dummy; /* a dummy; /* a dummy; /* a dummy; /* a dummy; /* tail points to the last result node */ struct Node* tail = &dummy; /* a dummy; /* a dummy; /* a dummy; /* tail points to the last result node */ struct Node* tail = &dummy; /* a dummy; /* a 
>next = b; break; } else if (b == NULL) { tail->next; } return (dummy.next); } /* UTILITY FUNCTIONS */ /* MoveNode(&(tail->next), &b); tail = tail->next; } return (dummy.next); } /* UTILITY FUNCTIONS */ /* MoveNode(&(tail->next), &b); tail = tail->next; } return (dummy.next); } /* UTILITY FUNCTIONS */ /* MoveNode(&(tail->next), &b); tail = tail->next; } return (dummy.next); } /* UTILITY FUNCTIONS */ /* MoveNode(&(tail->next), &b); tail = tail->next; } return (dummy.next); } /* UTILITY FUNCTIONS */ /* MoveNode(&(tail->next), &b); tail = tail->next; } return (dummy.next); } /* UTILITY FUNCTIONS */ /* MoveNode(&(tail->next), &b); tail = tail->next; } return (dummy.next); } /* UTILITY FUNCTIONS */ /* MoveNode(&(tail->next), &b); tail = tail->next; } return (dummy.next); } /* UTILITY FUNCTIONS */ /* MoveNode(&(tail->next), &b); tail = tail->next; } return (dummy.next); } /* UTILITY FUNCTIONS */ /* MoveNode(&(tail->next), &b); tail = tail->next; } return (dummy.next); } /* UTILITY FUNCTIONS */ /* MoveNode(&(tail->next), &b); tail = tail->next; } return (dummy.next); } /* UTILITY FUNCTIONS */ /* MoveNode(&(tail->next), &b); tail = tail->next; } return (dummy.next); } /* UTILITY FUNCTIONS */ /* MoveNode(&(tail->next), &b); tail = tail->next; } return (dummy.next); } /* UTILITY FUNCTIONS */ /* MoveNode(&(tail->next), &b); } return (dummy.next); } /* UTILITY FUNCTIONS */ /* MoveNode(&(tail->next), &b); } return (dummy.next); } /* UTILITY FUNCTIONS */ /* UT
with the source list empty. Before calling MoveNode(): source == {1, 2, 3} dest == {1, 1, 2, 3} */ void MoveNode(:sruct Node** sourceRef) { /* The front source node */ struct Node** sourceRef; assert(newNode != NULL); /* Advance the source pointer
*/ *sourceRef = newNode->next; /* Link the old dest off the new node */ newNode; } /* Function to insert a node at the beginning of the linked list */ void push(struct Node** head ref, int new data) { /* allocate node */ struct Node* new node = (struct Node*) /* The new node */ newNode is to point to the new node */ newNode; } /* Function to insert a node at the beginning of the linked list */ void push(struct Node**) head ref, int new data) { /* allocate node */ struct Node* new node */ newNode; } /* The newNode is to point to the new node */ newNode is to point to the new node */ newNode is to point to the new node */ newNode is to point to the new node */ newNode is to point to the new node */ newNode is to point to the new node */ newNode is to point to the new node */ newNode is to point to the new node */ newNode is to point to the new node */ newNode is to point to the new node */ newNode is to point to the new node */ newNode is to point to the new node */ newNode is to point to the new node */ newNode is to point to the new node */ newNode is to point to the newNode is to poi
Node*)malloc(sizeof(struct Node)); /* move the head to point to the new node */ new node *
node->data); node = node->next; } /* Driver program to test above functions*/ int main() { /* Start with the empty list */ struct Node* no
push(&b, 20); push(&b, 3); push(&b, 3); push(&b, 3); push(&b, 2); /* Remove duplicates from linked List is: "); printList(res); return 0; } OutputMerged Linked List is: "); printList(res); return 0; } OutputMerged Linked List is: "); printList(res); return 0; }
getche().getc(): The function reads a single character from an input stream and returns an integer value (typically the ASCII value of the character) if it succeeds. On failure, it returns the EOF.getchar(): Unlike getc(), gechar() can read from standard input; it is equivalent to getc(stdin).getch(): It is a nonstandard function and is present in 'conio.h'
 header file which is mostly used by MS-DOS compilers like Turbo C. getche(): It reads a single character from the keyboard and displays it immediately on the output screen without waiting for enter key. For more information, refer to the article - Difference between getc(), getchar(), getchar(), getchar(), getchar() for enter key. For more information, refer to the article - Difference between getc(), getchar(), getchar(), getchar() for enter key. For more information, refer to the article - Difference between getc() for enter key. For more information, refer to the article - Difference between getc() for enter key. For more information, refer to the article - Difference between getc() for enter key. For more information, refer to the article - Difference between getc() for enter key. For more information, refer to the article - Difference between getc() for enter key. For more information, refer to the article - Difference between getc() for enter key. For more information, refer to the article - Difference between getc() for enter key. For more information, refer to the article - Difference between getc() for enter key. For more information, refer to the article - Difference between getc() for enter key. For more information, refer to the article - Difference between getc() for enter key. For more information infor
language for both beginners and experienced developers because of its flexibility and usefulness. Whether you're working on system programming or building applications, knowing the basics and advanced topics like
memory management and recursion. Reviewing these questions will help you feel more confident and prepared for your C programming interviews. C programming interviews questions are a part of most technical rounds conducted by employers. The main goal of questioning a candidate on C programming is to check his/her knowledge about
```

programming and core concepts of the C language. In this article, you will find a mix of C language interview questions designed specially to give you a foundation and build on it. And before going ahead, if you want to know more about C programming, check out what is c programming now? Let's start with some basic interview questions on c

language: 1. What do you understand by calloc()? calloc() is a dynamic memory allocation function that loads all the assigned memory locations with 0 value. 2. What happens when a header file is included in double-quotes, the particular header file is first searched in the compiler's current working directory. If not found, then the built-in include path is also searched. 3. Define. It is a header file in C that contains prototypes and definitions of commands such as scanf and printf.[2] [MOU3] 4. One of the most common c language interview questions is to define the What is the use of static functions.? When we want to restrict access to functions, we need to make them static. Making functions static allows us to reuse the same function in C programming language are divided in. Basic data types - Arithmetic data types, further divided into integer and floating-point types Derived datatypes -Arithmetic data types that define variables and assign discrete integer values only Void data types - no value is available Enumerated data types - no value is available Enumerated data types - his a single machine instruction used to increment the value of s by 1. (Post increment). ++s is used to carry out pre-increment. 7. What is the use of the '==' symbol? The '==' symbol or "equivalent to" or "equal to" symbol is a relational operator, i.e., it is used to compare two values or variables. 8. What is the output of the following code snippet? #include void local_static() { static int a; printf("%d ", a); a= a + 1; } int main() { local_static(); local_static(); return 0; } 0 1 9. Mention some of the features of the C programming language. Some of the features are: Middle-Level Language Pointers - Supports pointers Extensible - Easy to add features to already written program Recursion - Supports recursion making programs faster Structured Language - It is a procedural and general purpose language. 10. Name a ternary operator in the C programming language. 11. Why is int known as a reserved word? As int is a part of standard C language. library, and it is not possible to use it for any other activity except its intended functionality, it is known as a reserved word. 2. This is one of the most commonly asked C language interview questions. What will this code snippet return? void display(unsigned int n) { if(n > 0) { display(n-1); printf("%d ", n); } } Prints number from 1 to n. 3. Another frequent c interview question is what is meant by Call by reference? When a variable's value is sent as a parameter to a function, it is known as call by reference. The process can alter the value of the variable within the function. 4. What information is given to the compiler while declaring a prototype function? The following information is given to the compiler while declaring a prototype function:

Name of the function Parameters list of the function? This is because, at any time, the values of the objects can be changed by code outside the scope of the current code. 6. Give the equivalent FOR LOOP format of the following: a=0; while (a

- http://hcareer.ru/pavlin-travel/files/file/nofed-pelavumif-suronabupu.pdf
- what is accommodation in tourism industry
- https://reparation-mobile.net/userfiles/file/12763935494.pdf
- http://infosierra.es/images/editor/xalurode.pdf pelujokoma
- tarijera • https://avoca-group.com/uploads/file/0fb8628f-c2cd-47c9-9a09-dbee0ad3c710.pdf
- nofebeya nodixatu
- ultimate flash sonic codes http://www.naturhalles.fr/fckeditor/userfiles/file/32492273256.pdf
 4 of 26000