

C# Interview questions

1. What is C#?

C# (C-Sharp) is a modern, general-purpose, object-oriented programming language (OOP) developed by Microsoft. It's primarily used on the Windows .NET framework, but can also be used on open source platforms.

C# is used in many areas of software development, including: Web and desktop app development, Game development, Microsoft Visual Studio, and Paint.NET

C# can be used to build a variety of things, including:

Windows client apps, Libraries, Components, Services, APIs, iOS apps, Android apps, Interoperability tools, Gaming systems, and Games.

2. What are the Features of C#

C# is a general-purpose programming language with many features, including:

1. Object-oriented: C# is a fully object-oriented language, which makes it easier to develop and maintain code.
2. Type safe: C# code can only access memory locations that it has permission to execute.
3. Interoperability: C# allows you to take advantage of existing investments in unmanaged code.
4. String interpolation: Introduced in C# 6.0, this feature allows you to embed expressions and variables directly within string literals.
5. Scalable and updateable: C# is a popular language for creating robust applications that can scale and be updated.
6. Fast: C# can make coding faster than other programming languages.
7. Easy to learn: C# is designed to be easy to learn, especially for programmers familiar with languages like Java and C++.

3.What are the difference between java and C#

| Feature | C# | Java |
|-----------------------------|--|--|
| Operator Overloading | C# supports operator overloading for multiple operators. | Java does not support operator overloading. |
| Runtime Environment | C# supports CLR (Common Language Runtime). | Java supports JVM (Java Virtual Machine). |
| API Control | C# APIs are controlled by an open-source community. | Java APIs are also controlled by open community processes support. |
| Public Classes | In C#, there can be many public classes inside a source code. | In Java there can be only one public class inside a source code otherwise there will be a compilation error. |
| Checked Exceptions | C# does not support Unix-based checked exceptions. In some cases checked exceptions are very useful for the smooth execution of the program. | Java supports both checked and unchecked exceptions. |
| Platform Dependency | C# is cross-platform and runs on both Windows & Unix-based systems. | Java is a robust and platform-independent language. The platform independence of Java is through JVM. |

4. Difference between C# and C++

Difference between C++ and C#(Sharp)

Below are some major differences between C++ and C#:

| Feature | C++ | C# |
|-----------------------------|---|--|
| Memory Management | In C++ memory management is performed manually by the programmer. If a programmer creates an object then he is responsible to destroy that object after the completion of that object's task. | In C# memory management is performed automatically by the garbage collector. If the programmer creates an object and after the completion of that object's task the garbage collector will automatically delete that object. |
| Platform Dependency | C++ code can be run on any platform. C++ is used where the application needed to directly communicate with hardware. | C# code is windows specific. Although Microsoft is working to make it global but till now the major system does not provide support for C#. |
| Multiple Inheritance | C++ support multiple inheritance through classes. Means that a class can extend more than one class at a time. | C# does not support any multiple inheritances through classes. |
| Bound Checking | In C++ bound checking is not performed by compiler. By mistake, if the programmer tries to access invalid array index then it will give the wrong result but will not show any compilation error. | In C# bound checking in array is performed by compiler. By mistake, if the programmer tries to access an invalid array index then it will give compilation error. |

| Feature | C++ | C# |
|----------------------------|--|---|
| Pointers | In C++ pointers can be used anywhere in the program. | In C# pointers can be used only in unsafe mode. |
| Language Type | C++ is a low level language. | C# is high level object oriented language. |
| Level of Difficulty | C++ includes very complex features. | C# is quite easy because it has the well-defined hierarchy of classes. |
| Application Types | C++ is typically used for console applications. | C# is used to develop mobile, windows, and console applications. |
| Compilation | C++ code gets converted into machine code directly after compilation. | C# code gets converted into intermediate language code after compilation. |
| Object Oriented | C++ is not a pure object-oriented programming language due to the primitive data types. | C# is a pure object-oriented programming language. |
| Access Specifiers | The access modifiers are public, private, protected. It does not contain internal & protected internal access modifiers. | In C# public, private, protected, internal & protected internal are used for access specifiers. |

| Feature | C++ | C# |
|---------------------------|--|--|
| Test Variable | In switch statement, the test variable can not be a string. | In switch statement, the test variable can be a string. |
| Control statement | It does not contain such extra flow control statement. | In addition to for, while and do while; it has another flow control statement called for each. |
| Function Pointers | It does have the concept of function pointers. | C# does have the concept of function pointers, but they are implemented differently compared to languages like C++. In C#, delegates serve as function pointers. |
| Binaries | In C++ , size of binaries is low and lightweight. | In C# , size of binaries is high because of overhead libraries. |
| Garbage Collection | C++ do not support garbage collection. | Garbage collection is supported by C# |
| Types of Projects | It is mainly used for such projects that focus on accessing the hardware and better performance. | It is mainly used in modern application development. |

5. Can multiple catch blocks be executed?

No, you cannot execute multiple catch blocks of the same type.

6. What is the difference between static, public, and void?

Public declared variables can be accessed from anywhere in the application. Static declared variables can be accessed globally without needing to create an instance of the class. Void is a type modifier which states the method and is used to specify the return type of a method in C#.

7. What are Jagged Arrays?

The Array which comprises elements of type array is called Jagged Array. The elements in Jagged Arrays can be of various dimensions and sizes.

8. What is the difference between out and ref parameters?

When an argument is passed as a ref, it must be initialized before it can be passed to the method. An out parameter, on the other hand, need not to be initialized before passing to a method.

9. What is the benefit of 'using' statement in C#?

The 'using' statement can be used in order to obtain a resource for processing before automatically disposing it when execution is completed.

10. Can "this" command be used within a static method?

No, we can't use "this" keyword inside a static method. "this" refers to current instance of the class. But if we define a method as static, class instance will not have access to it, only CLR executes that block of code. Hence we can't use "this" keyword inside static method.

11. Differentiate between Break and Continue Statement.

Continue statement - Used in jumping over a particular iteration and getting into the next iteration of the [loop](#).

Break statement - Used to skip the next statements of the current iteration and come out of the loop.

12. List the different types of comments in C#.

The different types of comments in C# are:

XML comments

Example -

```
/// example of XML comment
```

Single Line comments

Example -

```
// example of single-line comment
```

Multi-line comments

Example -

```
/* example of an  
multiline comment */
```

13. Explain the four steps involved in the C# code compilation.

Four steps of code compilation in C# include -

1. Source code compilation in managed code.
2. Newly created code is clubbed with assembly code.
3. The Common Language Runtime (CLR) is loaded.
4. Assembly execution is done through CLR.

14. Discuss the various methods to pass parameters in a method.

The various methods of passing parameters in a method include -

1. Output parameters: Lets the method return more than one value.
2. Value parameters: The formal value copies and stores the value of the actual argument, which enables the manipulation of the formal parameter without affecting the value of the actual parameter.
3. Reference parameters: The memory address of the actual parameter is stored in the formal argument, which means any change to the formal parameter would reflect on the actual argument too.

15. Name all the C# access modifiers.

The C# access modifiers are -

1. Private Access Modifier - A private attribute or method is one that can only be accessed from within the class.
2. Public Access Modifier - When an attribute or method is declared public, it can be accessed from anywhere in the code.
3. Internal Access Modifier - When a property or method is defined as internal, it can only be accessible from the current assembly point of that class.
4. Protected Access Modifier - When a user declares a method or attribute as protected, it can only be accessed by members of that class and those who inherit it.

16. Mention the important IDEs for C# development provided by Microsoft.

The following IDEs' are useful in C# development -

1. MonoDevelop
2. Visual Studio Code (VS Code)
3. Browxy
4. Visual Studio Express (VSE)
5. Visual Web Developer (VWD)

17. Why do we use C# language?

Below are the reasons why we use the C# language -

1. C# is a component-oriented language.
2. It is easy to pass parameters in the C# language.
3. The C# language can be compiled on many platforms.
4. The C# language follows a structured approach.
5. It is easy to learn and pick up.
6. The C# language produces really efficient and readable programmes.

18. What is meant by Unmanaged or Managed Code?

1. Managed code is the code which is managed by the CLR(Common Language Runtime) in .NET Framework. Whereas the Unmanaged code is the code which is directly executed by the operating system.
2. In simple terms, managed code is code that is executed by the CLR (Common Language Runtime). This means that every application code is totally dependent on the .NET platform and is regarded as overseen in light of it. Code executed by a runtime programme that is not part of the .NET platform is considered unmanaged code. Memory, security, and other activities related to execution will be handled by the application's runtime.

19. What is meant by a Partial Class?

1. Technically, a partial class is a single class that can be divided into different files, all within the same assembly. Due to assembly boundaries, you can't have a part of a class in one assembly and another part in a different assembly
2. A [partial class](#) effectively breaks a class's definition into various classes in the same or other source code files. A class definition can be written in numerous files, but it is compiled as a single class at runtime, and when a class is formed, all methods from all source files can be accessed using the same object. The keyword 'partial' denotes this.

20. What is the difference between read-only and constants?

During the time of compilation, constant variables are declared as well as initialized. It's not possible to change this particular value later. On the other hand, read-only is used after a value is assigned at run time.

21. What are reference types and value types?

A value type holds a data value inside its memory space. Reference type, on the other hand, keeps the object's address where the value is stored. It is, essentially, a pointer to a different memory location.

22. What are User Control and Custom Control?

Custom Controls are produced as compiled code. These are easy to use and can be added to the toolbox. Developers can drag and drop these controls onto their web forms. User Controls are almost the same as ASP include files. They are also easy to create. User controls, however, can't be put in the toolbox. They also can't be dragged and dropped from it.

23. What are sealed classes in C#?

When a restriction needs to be placed on the class that needs to be inherited, sealed classes are created. In order to prevent any derivation from a class, a sealed modifier is used. Compile-time error occurs when a sealed class is forcefully specified as a base class.

24. Is it possible for a private virtual method to be overridden?

A private virtual method cannot be overridden as it can't be accessed outside the class.

25. Describe the accessibility modifier "protected internal".

Variables or methods that are Protected Internal can be accessed within the same assembly as well as from the classes which have been derived from the parent class.

26. What are the differences between System.String and System.Text.StringBuilder classes?

System.String is absolute. When a string variable's value is modified, a new memory is assigned to the new value. The previous memory allocation gets released.

System.StringBuilder, on the other hand, is designed so it can have a mutable string in which a plethora of operations can be performed without the need for allocation of a separate memory location for the string that has been modified.

27. What's the difference between the `System.Array.CopyTo ()` and `System.Array.Clone ()` ?

In the `Clone ()` method, a new array object is created, with all the original Array elements using the `CopyTo ()` method. Essentially, all the elements present in the existing array get copied into another existing array.

28. How can the Array elements be sorted in descending order?

You can use the `Sort()` methods and then `Reverse()` method.

29. What is the difference between `Dispose()` and `Finalize()` methods?

`Dispose()` is used when an object is required to release any unmanaged resources in it. `Finalize()`, on the other hand, doesn't assure the garbage collection of an object even though it is used for the same function.

30. What are circular references?

When two or more resources are dependent on each, it causes a lock condition, and the resources become unusable. This is called a circular reference.

31. What are generics in C# .NET?

In order to reduce code redundancy, raise type safety, and performance, generics can be used in order to make code classes that can be reused. Collection classes can be created using generics.

32. What is an object pool in .NET?

A container that has objects which are ready to be used is known as an object pool. It helps in tracking the object which is currently in use and the total number of objects present in the pool. This brings down the need for creating and re-creating objects.

33. List down the most commonly used types of exceptions in .NET

Commonly used types of exceptions in .NET are:

`ArgumentException`

ArithmeticException

DivideByZeroException

OverflowException

InvalidCastException

InvalidOperationException

NullReferenceException

OutOfMemoryException

StackOverflowException

34. What are Custom Exceptions?

In some cases, errors have to be handled according to user requirements. Custom exceptions are used in such cases.

35. What are delegates?

Delegates are essentially the same as function [pointers in C++](#). The main and only difference between the two is delegates are type safe while function pointers are not. Delegates are essential because they allow for the creation of generic type-safe functions.

36. How do you inherit a class into another class in C#?

In C#, colon can be used as an inheritance operator. You need to place a colon and follow it with the class name.

37. Why can't the accessibility modifier be specified for methods within the interface?

In an interface, there are virtual methods which do not come with method definition. All the methods present are to be overridden in the derived class. This is the reason they are all public.

38. How can we set the class to be inherited, but prevent the method from being overridden?

To set the class to be inherited, it needs to be declared as public. The method needs to be sealed to prevent any overrides.

39. What happens if the method names in the inherited interfaces conflict?

A problem could arise when the methods from various interfaces expect different data. But when it comes to the compiler itself, there shouldn't be an issue.

40. What is the difference between a Struct and a Class?

Structs are essentially value-type variables, whereas classes would be reference types.

41. How to use nullable types in .Net?

When either normal values or a null value can be taken by value types, they are called nullable types.

42. How can we make an array with non-standard values?

An array with non-default values can be created using `Enumerable.Repeat`.

43. What is the difference between "is" and "as" operators in c#?

An "is" operator can be used to check an object's compatibility with respect to a given type, and the result is returned as a Boolean. An "as" operator can be used for casting an object to either a type or a class.

44. What is a multicast delegate?

Multicast delegate is when a single delegate comes with multiple handlers. Each handler is assigned to a method.

45. What are indexers in C# .NET?

In C#, indexers are called smart arrays. Indexers allow class instances to be indexed in the same way as arrays do.

46. What is the distinction between "throw" and "throw ex" in .NET?

"Throw" statement keeps the original error stack. But "throw ex" keeps the stack trace from their throw point.

47. What are C# attributes and its significance?

C# gives developers an option to define declarative tags on a few entities. For instance, class and method are known as attributes. The information related to the attribute can be retrieved during runtime by taking the help of Reflection

48. In C#, how do you implement the singleton design pattern?

In a singleton pattern, a class is allowed to have only one instance, and an access point is provided to it globally.

49. What's the distinction between directcast and ctype?

If an object is required to have the run-time type similar to a different object, then DirectCast is used to convert it. When the conversion is between the expression as well as the type, then CType is used.

50. Is C# code managed or unmanaged code?

C# is a managed code as the runtime of Common language can compile C# code to Intermediate language.

51. What is a Console application?

An application that is able to run in the command prompt window is called a console application.

52. What are namespaces in C#?

Namespaces allow you to keep one set of names that is different from others. A great advantage of namespace is that class names declared in one namespace don't clash with those declared in another namespace.

53. Difference between SortedList and SortedDictionary in C#.

SortedList is a collection of value pairs sorted by their keys. SortedDictionary is a collection to store the value pairs in the sorted form, in which the sorting is done on the key.

54. What is Singleton design pattern in C#?

Singleton design pattern in C# has just one instance that gives global access to it.

55. What is tuple in C#?

Tuple is a data structure to represent a data set that has multiple values that could be related to each other.

56. What are Events?

An event is a notice that something has occurred.

57. What is the Constructor Chaining in C#?

With Constructor Chaining, an overloaded constructor can be called from another constructor. The constructor must belong to the same class.

58. What is a multicasting delegate in C#?

Multicasting of delegates helps users to point to more than one method in a single call.

59. What are Accessibility Modifiers in C#?

Access Modifiers are terms that specify a program's member, class, or datatype's accessibility.

60. What is a Virtual Method in C#?

In the parent class, a virtual method is declared that can be overridden in the child class. We construct a virtual method in the base class using the virtual keyword, and that function is overridden in the derived class with the Override keyword.

61. What is Multithreading with .NET?

Multi-threading refers to the use of multiple threads within a single process. Each thread here performs a different function.

62. In C#, what is a Hash table class?

The Hash table class represents a collection of key/value pairs that are organized based on the hash code of the key.

63. What is LINQ in C#?

LINQ refers to Language Integrated Query. It provides .NET languages (like C#) the ability to generate queries to retrieve data from the data source.

64. Why can't a private virtual procedure in C# be overridden?

Private virtual methods are not accessible outside of the class.

65. What is File Handling in C#?

File handling includes operations such as creating the file, reading from the file, and appending the file, among others

66. Why are Async and Await used in C#?

Asynchronous programming processes execute independently of the primary or other processes. Asynchronous methods in C# are created using the Async and Await keywords.

67. What are I/O classes in C#?

In C#, the System.IO namespace contains multiple classes that are used to conduct different file operations such as creation, deletion, closure, and opening.

68. What is an Indexer in C#?

An indexer is a class property that allows you to access a member variable of another class using array characteristics.

69. What is Thread Pooling in C#?

In C#, a Thread Pool is a group of threads. These threads are used to do work without interfering with the principal thread's operation.

70. What exactly do you mean by regular expressions in C#?

A regular expression is a pattern that can be used to match a set of input. Constructs, character literals, and operators are all possible.

71. Can you write a code on Boxing and Unboxing in C#?

Ans: The methods of boxing and unboxing are used for type conversions. Boxing is converting from a value type to a reference type. Boxing is an implicit conversion. Here's an example of C# boxing.

```
// Boxing
int num1 = 1;
Object obj = num1;
Console.WriteLine(num1);
Console.WriteLine(obj);
```

Unboxing is the process of converting from a reference type to a value type. Here's a C# example of unboxing.

```
// Unboxing
Object obj2 = 1;
int num2 = (int) obj;
```

```
Console.WriteLine(num2);
```

```
Console.WriteLine(obj);
```

72. In C#, What is the difference between a struct and a class?

Ans: Class and struct are both user-defined data types, but they differ significantly:

| Struct | Class |
|---|--|
| The struct is a value type in C# inherited from System.Value Type. | The class is a reference type in C# inherited from the System.Object Type. |
| Struct is usually used for smaller amounts of data. | Classes are usually used for large amounts of data. |
| Struct can't be inherited from other types. | Classes can be inherited from other classes. |
| A structure can't be abstract. | A class can be an abstract type. |
| No need to create an object with a new keyword. Do not have permission to create any default constructor. | We can create a default constructor. |

73. What exactly is cohesion?

Ans: In OOPS, we write code in modules. Each module is responsible for a specific task. Cohesion measures how closely a module's responsibilities are related.

Greater cohesion is always preferable. Benefits of increased cohesion include:

- Module maintenance is improved.
- Improve reusability.

74. Why are strings immutable in C#?

Ans: String values are immutable, which means they cannot be changed once they have been created. Any change to a string value creates a new string instance, resulting in inefficient memory use and extraneous garbage collection. The System is malleable. When string values change, the Text.StringBuilder class should be used.

75. What are the distinctions between Array and Array List in C#?

Ans: Arrays store values or elements of the same data type, whereas array lists store values of various data types.

Arrays will use a fixed length, whereas array lists will not.

76. What is the distinction between System.String and System.Text.StringBuilder?

Ans: Systems.Strings are unchangeable. When we change the value of a string variable, we allocate new memory to the new value and release the previous memory allocation. System.Text.StringBuilder was created with the concept of a mutable string in mind, allowing a variety of operations to be performed without the need for a separate memory location for the modified string.

77. What exactly is reflection in C#?

Ans: During runtime, C# reflection extracts metadata from data types.

To implement reflection in the [.Net](#) framework, simply use the System.Reflection namespace in your program to retrieve the type, which can be any of the following:

Assembly

ConstructorInfo

MemberInfo

ParameterInfo

FieldInfo

EventInfo

PropertyInfo

78. What is the purpose of 'Data Type Conversion' in C#?

Ans: The purpose of data type conversion is to avoid situations of runtime error during data type change or conversion.

79.What is the GAC, and where can I find it?

Ans: The Global Assembly Cache is abbreviated as the GAC. Shared assemblies are stored in the GAC, allowing applications to share assemblies rather than having them distributed with each application. Thanks to versioning, multiple assembly versions can exist in the GAC—applications can specify version numbers in the config file. To manage the GAC, use the gacutil command line tool.

80. What is a circular reference in C#?

Ans: This is a situation in which multiple resources depend on each other, resulting in a lock condition and unused resources.

81.What is Garbage collection in c#?

The garbage collector (GC) is in charge of memory allocation and release. The garbage collector is a memory manager that runs in the background. You don't need to know how to allocate and release memory or how to manage the lifetimes of the things that use it.

82. What is the distinction between the methods Finalize() and Dispose()?

Ans: When we want an object to release unmanaged resources, we call dispose(). Finalize(), on the other hand, serves the same purpose but does not guarantee garbage collection of an object.

83. What exactly are Anonymous Types in C#?

Ans: Anonymous types enable us to create new types without having to define them. This method defines read-only properties in a single object without explicitly defining each type. Type is generated by the compiler and is only available for the current block of code. The compiler also infers the type of properties.

C# programming Questions

1. write a code to reverse a string

```
Main.cs
1  using System;
2
3  class Program
4  {
5      static void Main()
6      {
7          Console.Write("Enter a String: ");
8          string originalString = Console.ReadLine();
9          string reverseString = string.Empty;
10         for (int i = originalString.Length - 1; i >= 0; i--)
11         {
12             reverseString += originalString[i];
13         }
14         Console.WriteLine($"Reversed String: {reverseString}");
15     }
16 }
17
```

2. Write a code to find the maximum and minimum of array

Main.cs

Share

Run

```
1 using System;
2 class Program
3 {
4     static void Main()
5     {
6         // Assume the first element is the maximum
7         int[] arr = { 14, 7, 25, 31, 10, 42 };
8         int max = arr[0];
9         for (int i = 1; i < arr.Length; ++i)
10        {
11            if (arr[i] > max)
12            {
13                max = arr[i];
14            }
15        }
16        Console.WriteLine($"Maximum value in the array: {max}");
17    }
18 }
19
```

Main.cs

Share

Run

```
1 using System;
2 class Program
3 {
4     static void Main()
5     {
6         // Assume the first element is the maximum
7         int[] arr = { 14, 7, 25, 31, 10, 42 };
8         int min = arr[0];
9         for (int i = 1; i < arr.Length; ++i)
10        {
11            if (arr[i] < min)
12            {
13                min = arr[i];
14            }
15        }
16        Console.WriteLine($"Maximum value in the array: {min}");
17    }
18 }
19
```

3.write a code to find the Fibonacci series

```
Main.cs
1 using System;
2 class Program
3 {
4     static void Main()
5     {
6         int firstNumber = 0, secondNumber = 1, nextNumber, numberOfElements;
7         Console.Write("Enter the number of elements to print: ");
8         numberOfElements = int.Parse(Console.ReadLine());
9
10        if (numberOfElements < 2)
11        {
12            Console.Write("Please enter a number greater than two.");
13        }
14        else
15        {
16            Console.Write(firstNumber + " " + secondNumber + " ");
17            for (int i = 2; i < numberOfElements; i++)
18            {
19                nextNumber = firstNumber + secondNumber;
20                Console.Write(nextNumber + " ");
21                firstNumber = secondNumber;
22                secondNumber = nextNumber;
23            }
24        }
25
26        Console.ReadKey();
27    }
28 }
29
```

4.write a code to find the factorial of a number

```
Main.cs
1 using System;
2 class Program
3 {
4     static void Main()
5     {
6         Console.Write("Enter a Number: ");
7         int number = int.Parse(Console.ReadLine());
8
9         long factorial = 1;
10        for (int i = 1; i <= number; i++)
11        {
12            factorial *= i;
13        }
14
15        Console.WriteLine($"Factorial of {number} is: {factorial}");
16    }
17 }
18
```

5.write a code to find a anagrams

Main.cs

```
1 using System;
2 using System.Linq;
3 class Program
4 {
5     public static bool AreAnagrams(string str1, string str2)
6     {
7         // Remove non-alphanumeric characters and convert to lowercase
8         str1 = new string(str1.Where(char.IsLetterOrDigit).ToArray()).ToLower();
9         str2 = new string(str2.Where(char.IsLetterOrDigit).ToArray()).ToLower();
10        // Sort the characters
11        char[] sortedStr1 = str1.ToCharArray();
12        char[] sortedStr2 = str2.ToCharArray();
13        Array.Sort(sortedStr1);
14        Array.Sort(sortedStr2);
15        // Compare the sorted strings
16        return sortedStr1.SequenceEqual(sortedStr2);
17    }
18
19    static void Main()
20    {
21        string input1 = "listen";
22        string input2 = "silent";
23
24        if (AreAnagrams(input1, input2))
25        {
26            Console.WriteLine($"{input1} and {input2} are anagrams!");
27        }
28        else
29        {
30            Console.WriteLine($"{input1} and {input2} are not anagrams.");
31        }
32    }
33 }
34
```

6. write a code to print star pattern in c#

```
Main.cs
1  using System;
2
3  class Program
4  {
5      static void Main(string[] args)
6      {
7          string series = string.Empty;
8          for (int i = 0; i <= 5; i++)
9          {
10             series += "*";
11             Console.WriteLine(series);
12         }
13
14         Console.ReadLine();
15     }
16 }
17
```

7.write a code to swap two numbers

```
Main.cs
1 using System;
2 class Program
3 {
4     static void Main()
5     {
6         int a = 5, b = 10;
7
8         Console.WriteLine($"Before swap: a = {a}, b = {b}");
9
10        a = a + b; // a = 15 (5 + 10)
11        b = a - b; // b = 5 (15 - 10)
12        a = a - b; // a = 10 (15 - 5)
13
14        Console.WriteLine($"After swap: a = {a}, b = {b}");
15    }
16 }
17
```

8.write a code to find palindrome

Main.cs



Share

```
1 using System;
2 class Program
3 {
4     static void Main()
5     {
6         Console.Write("Enter a number: ");
7         int num = int.Parse(Console.ReadLine());
8         string original = num.ToString();
9         char[] reversed = original.ToCharArray();
10        Array.Reverse(reversed);
11        string reversedString = new string(reversed);
12        if (original == reversedString)
13        {
14
15            Console.WriteLine($"{num} is a palindrome!");
16        }
17        else
18        {
19            Console.WriteLine($"{num} is not a palindrome.");
20        }
21    }
22 }
```