**All the answers should be written inside Program.cs.**

* **Short-answer (theory) questions → should be written as comments (// or /\* ... \*/).**
* **Coding questions → should each be placed in a separate class, with the solution implemented as a function.**

**01 Introduction to C# and Data Types – Questions**

1. What type would you choose for the following “numbers”?  
     
   * A person’s telephone number
   * A person’s height
   * A person’s age
   * A person’s gender (Male, Female, Prefer Not To Answer)
   * A person’s salary
   * A book’s ISBN
   * A book’s price
   * A book’s shipping weight
   * A country’s population
   * The number of stars in the universe
   * The number of employees in each of the small or medium businesses in the United Kingdom (up to about 50,000 employees per business)
2. What are the differences between value type and reference type variables?  
    What is boxing and unboxing?
3. What is meant by the terms **managed resource** and **unmanaged resource** in .NET?
4. What is the purpose of the Garbage Collector in .NET?

**Controlling Flow and Converting Types – Questions**

1. What happens when you divide an int variable by 0?
2. What happens when you divide a double variable by 0?
3. What happens when you overflow an int variable (assign a value beyond its range)?
4. What is the difference between x = y++; and x = ++y;?
5. What is the difference between break, continue, and return when used inside a loop statement?
6. What are the three parts of a for statement and which of them are required?
7. What is the difference between the = and == operators?
8. Does the following statement compile? for ( ; true; ) ;
9. What interface must an object implement to be enumerated by the foreach statement?

**Coding**：

1. How can we find the minimum and maximum values, as well as the number of bytes, for the following data types: sbyte, byte, short, ushort, int, uint, long, ulong, float, double, and decimal?

2. Write a method in C# called **FizzBuzz** that takes an integer num and prints numbers from 1 up to num, but:

* Print **Fizz** if the number is divisible by 3.
* Print **Buzz** if the number is divisible by 5.
* Print **FizzBuzz** if the number is divisible by both 3 and 5.
* Otherwise, print the number itself.

3. What will happen if this code executes?

int max = 500;

for (byte i = 0; i < max; i++)

{

    Console.WriteLine(i);

}

4. **Two Sum** Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target.

* You may assume that each input would have exactly one solution.
* You may not use the same element twice.
* You can return the answer in any order.