Concept Questions:

1. What type would you choose for the following “numbers”?

| | |
|---|---|
| A person’s telephone number: | int |
| A person’s height: | float |
| A person’s age: | int |
| A person’s gender (Male, Female, Prefer Not To Answer): | char/string |
| A person’s salary: | decimal |
| A book’s ISBN | int/string |
| A book’s price | decimal |
| A book’s shipping weight | float |
| A country’s population | int |
| The number of stars in the universe | long |
| The number of employees in each of the small or medium businesses in the United Kingdom (up to about 50,000 employees per business) | int |

2. What are the difference between value type and reference type variables?

- Value type variables can directly contain their data while reference type variables store references to their data known as objects.
- Each value type has its own copy of data, and operations on one variable will not affect other.
- Two different reference type variables can refer to the same object(overloading), and operations on one variable can affect other.

3. What happens when you divide an int variable by 0?

It will throw a DivideByZeroException exception. An error will occur as: Division by constant zero.

4. What happens when you divide a double variable by 0?

It will not throw an exception. It results in positive infinity, negative infinity, or not a number(NaN), according to the rules of IEEE 754 arithmetic.

5. What happens when you overflow an int variable, that is, set it to a value beyond its range?

Under different execution situation, we will get different results.
Under a checked context, an exception will be thrown.
Under an unchecked context, the most significant bits of the result will be discarded and the execution continues.

6. What is the difference between x = y++; and x = ++y;?

x=y++ is postfix. In this case, x will become y, then y will become y+1. Postfix processes the data/command before it adds the value.
x=++y is prefix. In this case, x will become y+1, then y will become y+1. Prefix will add the value before processing the data/command.

7. What is the difference between break, continue, and return when used inside a loop statement?
All these statements are jump statements.
Break: The break statement terminates the closest enclosing iteration statement or switch statement. In loop, break will exit or escape the for loop.
Continue: The continue statement starts a new iteration of the closest enclosing iteration statement. In loop, it begins the next iteration of the loop.
Return: The return statement terminates execution of the function in which it appears and returns control to the caller. In loop, return will exit the entire method that is currently executing.

8. What are the three parts of a for statement and which of them are required?
A initialization, a condition, and a iterator. All three statements are optional but will bring different result based on lack of certain statement.
If we only define the condition, the for loop will act like a while loop until condition meet.
If we do not define anything, the loop will run forever, like a infinite loop.

LeetCode Questions:

Q1. Two Sum

```
public class Solution {
    public int[] TwoSum(int[] nums, int target) {
        Dictionary<int, int> map = new Dictionary<int, int>();
        int[] result = new int[2];
        for (int i = 0; i < nums.Length; i++) {
            int complement = target - nums[i];
            if (map.ContainsKey(complement)) {
                result[0] = map[complement];
                result[1] = i;
                return result;
            }
            map[nums[i]] = i;
        }
        return result;
    }
}
```

Q9. Palindrome Number

```
public class Solution {
    public bool IsPalindrome(int x) {
        string xs = x.ToString();
        for (int i = 0; i<xs.Length/2; i++){
            if(xs[i] != xs[xs.Length-1-i])
            {
                return false;
            }
        }
        return true;
    }
}
```

Q217. Contains Duplicates—Solution 1

```
public class Solution {
    public bool ContainsDuplicate(int[] nums) {
        HashSet<int> seen = new HashSet<int>();
        for(int i = 0;i<nums.Length; i++){
            if(seen.Contains(nums[i])){
                return true;
            }
            seen.Add(nums[i]);
        }
        return false;
        }
}
```

OR

Q217. Contains Duplicates—Solution 2

Q412. Fizz Buzz

```
public class Solution {
    public IList<string> FizzBuzz(int n) {
        List<string> k = new();

        for (int i = 1; i <= n; i++)
        {
            if (i % 3 == 0 && i % 5 == 0) {
                k.Add("FizzBuzz");
            } else if (i % 3 == 0) {
                k.Add("Fizz");
            } else if (i % 5 == 0) {
                k.Add("Buzz");
            } else {
                k.Add($"{i}");
            }
        }
        return k;
    }
}
```