Gamigo Programming Test C#

Please send back your solutions as C# files attached to your email. Do not paste your answers into this document or into the body of your email.

There is no set time limit. However, please let us know the approximate amount of time you spent on the test. You may use any reference materials (books, internet sites, etc.). We strongly recommend that you compile and test your answers. If you write any test code, please include it with your answers.

Write your answers in C# assuming they will become part of a large game codebase. Your code will be called frequently during gameplay and will be maintained and extended by the entire programming team.

1. Write a method that takes an unsigned integer as input and returns true if all the digits in the base 10 representation of that number are unique.

bool AllDigitsUnique(unsigned int value)

Example:

AllDigitsUnique(48778584) returns false

AllDigitsUnique(17308459) returns true

2. Write a method that modifies an input string, sorting the letters according to a sort order defined by a second string. You may assume that every character in the input string also appears somewhere in the sort order string. Make your method as fast as possible for long input strings.

```
void SortLetters(byte[] inputAndOutput, byte[] sortOrder)
```

Example:

```
string inputStr = “trion world network”;
string sortStr = " oinewkrtdl";

byte[] inputAndOutputBytes = System.Text.Encoding.ASCII.GetBytes(inputStr);
byte[] sortBytes = System.Text.Encoding.ASCII.GetBytes(sortStr);

SortLetters(inputAndOutputBytes, sortBytes);

string outputStr = System.Text.Encoding.ASCII.GetString(inputAndOutputBytes);
```

would set the output string to " oooinnewwkrrrttdl".

3. Imagine a maze of interconnected rooms:

- Each room has a unique name and can be connected to between zero and four other rooms through doors named north, south, east, and west.

- Rooms are not necessarily spatially coherent. If A is north of B, B might not be south of A.

- Doors are not necessarily bidirectional. If A can be reached from B, B might not be reachable from A.

- Rooms might connect to themselves.

3a. Write a class or struct declaration that shows the structure of a Room. Include a constructor (for classes) or initialization method (for structs).

3b. Write a method to connect a new Room to an existing Room.

3c. Write a method to determine if any path exists between a starting room and an ending room with a given name. You may add additional members and methods to your Room class or struct as needed. You do not need to worry about finding the shortest path. It is only necessary to determine if any path exists.

Your Room class/structure should use the following interface in your answer:

```
public interface IPathable
{
  bool PathExistsTo(string endingRoomName).
}
```

4. Design an elevator controller for a building of ten floors (including the ground floor).

Your ElevatorController class should implement the IElevatorController interface below.
Your ElevatorMotor class should implement the IElevatorMotor interface below and needs to be referenced by the code you submit.

Design your controller to achieve the overall-most-efficient system. The definition of overall-most-efficient is up to you.

```
public interface IElevatorMotor
{
  Direction CurrentDirection { get; }
  int CurrentFloor { get; }

  event Action<int> ReachedFloor;

  void GoToFloor(int floor);
}

public interface IElevatorController
{
  void SummonButtonPushed(int floor, Direction d);
  void FloorButtonPushed(int floor);

  event Action<int> ReachedSummonedFloor;
  event Action<int> ReachedDestinationFloor;
}
```