BT Studios Programmer Test

This test is meant to simulate workplace type problem solving and hard challenges. Take your time and research or use Google as much as you like, as you would in the workplace.

You are encouraged to work towards answers that show brevity and efficiency.

Find a care package emailed to you with the files you need. Email all code and answers to questions to: chris@btstudios.net

If there are any errors in the test or files preventing compiling, or the web pages don't load etc. feel free to contact Chris via email.

Part 1: C# writing and problem solving: “The worst videogame ever made”

Go to youtube and search for “Sword of Sodan long play [387]”. Within the first 20 seconds of the video you will see a title screen effect. Recreate this retro “programmer art” effect as closely as you can in C# using whatever means you like to bring it to the screen (only interested in the code that renders the effect).

You will be judged on the efficiency of the algorithm. You will find the needed art assets in the zip.

Part 2: Ability to Solve errors "debugging your way to freedom"

Debugging is a key skill to developing code for a large and complex project.

In finding a bug, a programmer might go through this sequence of progressively more radical yet effective debugging techniques to isolate a problem:

1: Re-read code to try to keyball the problem.

2: Print out variables to clarify data.

3: Examine the call stack to clarify execution.

4: Use a debugger to watch data, execution or fire breakpoints - to explicitly clarify data and execution.

Question 1: In the worse case there is one even more radical debugging technique used to isolate a problems, what is it? Explain.

Question 2: As a project grows larger and the code base more extensive, the amount of bugs and their severity grows. Explain what causes this phenomenon.

Question 3: Can you think of any solution(s) to the above problem? Explain how you would execute this.

OPTIONAL BONUS QUESTION

Part 3: Code Reading / Understanding / Amending: "Goto statements? BAD FORM"

In the zip you will find an executable for a grid based logic game. Run the game to test it.(It should work on windows 7 / 10 (or under wine on mac)).

Gameplay: Chain together matching powers of 2, and successive increasing tiles with your mouse, press enter on the keyboard to input.

Valid examples:

16,16
2,2,4,8
8,8,16,WILD,64

To win: score >= 5000 out of 5000

In the zip you will also find the code for the game. The only relevant code to you is everything above line 140, however it may assist understanding to read the rest.

A lot of times in a job setting you might have to read obfuscated code and make sense of it. You may also have to work with programming languages outside your comfort zone.

A1:

Look at the function "**validateSequence**()", the logic in this function is entirely pure, or side effect free. (if you are not familiar with c++ you will find validateSequence.cs (compare them to discover nearly identical logic))

Question 1: Explain the merits of a pure function.
Question 2: Explain what this function does.v
Question 3: Add in missing comments for lines on this function without comments.

A2:

Question 1: Read, understand, and add comments to the function isValidTile().

Question 2: It is generally discouraged to use goto statements in modern code, explain why.

Question 3: Does the common wisdom about goto statements apply here? Explain.

A3:

Read, understand, and add comments explaining the other 2 functions above line 130.

A4:

Notice the prototype for the function WinsPossible(). This code that is compiled into the exe is missing. You are tasked with writing the code (in either c++ or equivalent c#). The code should:

-return a bool showing if a win is possible in the current game state. -fill a string with the sequence of moves to complete the win.

HINT: Check out the gamestate structure in the header and notice the existence of its data and copy constructor.