

Application Take Home Test

1 Objectives

The primary objective of this take home test is to evaluate your skill as a programmer and software engineer. Keep that in mind as you read the rest of this document and as you work on the code.

Estimated completion time is 4 to 8 hours for the bare minimum project.

All code shall be written from scratch.

Here are some of the evaluation criteria used:

- A) Familiarity with C/C++
 - a. Is good use made of the language?
 - b. Are good programming practices followed?
- B) Software design skills
 - a. Is the code modular?
 - b. Is the functionality encapsulated neatly within each module?
 - c. Are the modules decoupled from one another?
- C) Documentation
 - a. Is the code commented well?
 - b. Is the code self-documenting?
- D) Organization
 - a. Is the project organized in a logical fashion?
 - b. Are GUIs laid out neatly or are the widgets scattered around?

2 Assignment

2.1 Circular Queue

You are to implement a circular queue in C. The target application is an 8-bit microcontroller.

2.1.1 Configurability

This module shall be configurable via preprocessor defines in the header file. Unless otherwise noted, all configurable parameters will require re-compilation.

The maximum size shall be configurable up to 256 elements.

The data type stored shall be easily modifiable via a typedef.

The module shall allow users to instantiate multiple queues.



2.1.2 Behavior

The queue shall not allow any data corruption.

No function call shall result in a memory overflow or crash.

Adding an element to a full queue results in no change to the contents of the queue.

Removing an element from an empty queue results in return of a default value.

2.2 Unit Tests

Now that your circular queue is complete, unit testing is required.

2.2.1 Framework

You are to develop a unit test application using Qt 5. The application shall have a GUI with a button that starts the unit test. You may add additional widgets to give insight into the testing process if you wish.

This application shall be in C++ even though the module is written in pure C.

The MinGW compiler shall be used.

The unit tests shall build and run on Windows 7 and / or Mac OS X.

2.2.2 Testing

Be thorough in your unit tests. This should be a small module so 100% test coverage is easily possible. Any documentation as to possible sources of bugs and how your testing covers edge cases.

2.2.3 Reusability

In a work environment, you will want to re-use the unit testing code for multiple modules and also be able to perform automated regression testing. Keep this in mind as you design your program.

2.2.4 Report

The unit test module shall output a report in HTML that presents the results of running the tests. This report can use a CSS framework such as Twitter Bootstrap, but be sure to include all of the necessary files in your submission.

3 Submission

All code shall be submitted by committing the code to your fork of the Github repository, then when you are ready, submit a pull request. Your submission will be evaluated once the pull request is received. If you do not hear back within 48 hours of submitting your pull request then please contact hiring@skywavedesign.com.

Please edit the README.txt to include (along with any other pertinent information) your contact information. You may also include a copy of your resume in the submission for reference.