Viable Alternatives: Practice Reading and Disassembling Object Code

In this assignment you will practice your skills analyzing the operation of a program for which you do not have the source code (the program is named `viable`). The program is designed to prompt a customer for some kind of feedback about a product or service. Refer to `disassemble.hex`, `disassemble.obj` and `disassemble.sections.txt` to help you answer the questions below. The contents of those three files were generated by the execution of three different programs:

	Filename	Generating Command
1.	`disassemble.hex`	`hexdump -C viable`
2.	`disassemble.obj`	`objdump -d -Mintel viable`
3.	`disassemble.sections.txt`	`readelf -S viable`

`viable` is implemented in an object-oriented paradigm. Central to the application is a `CustomerFeedback` class. The app's developers used various derived classes to implement additional functionality for different types of customer feedback. In particular, the developers created a `QuantitativeFeedback` class derived from the `CustomerFeedback` class for those situations where a user is looking for quantitative feedback from customers.

To complete the assignment, you may want to refer to some of the tools that we have learned in the course. For a refresher on the most important ones, refer to the SAID. You may also want to refer to documentation about the ELF file format.

1. At what address does the `.text` section start *when the program is running*?

_____

2. At what offset (in bytes) does the `.rodata` section start in `viable` on the disk?

_____

3. At what address does the `.text` section end *when the program is running*?

_____

4. What is the *demangled* name of the function `call`'d at address `0x401232`?

_____

5. What is the value of `rdi` at the time the function is `call`'d at `0x401232`?

_____

6. Using the information that you gathered to answer questions (4) and (5), what does the value of `rdi` represent

7. Assume that the constructor of the `QuantitativeFeedback` class takes one parameter, a prompt that the software will use to solicit customer feedback (the parameter's type is `char *`). The constructor assigns the value of that parameter to a member variable named `m_question`. Further, assume that the function `display_feedback` takes one parameter, a pointer to an instance of `QuantitativeFeedback`, and performs only one action: it prints the value of it's parameter's `m_question` member variable. What does `viable` print?

8. The `CustomerFeedback` class contains two pure virtual member functions. `QuantitativeFeedback` implements each of those.
    1. Where is the virtual table (vtable) for the `CustomerFeedback` class?

_____

    2. Where is the vtable for the `QuantitativeFeedback` class?

_____

    3. The pointer to which of two member functions comes first in the vtable?

_____

9. Describe the process by which you arrived at your answers for Question 8. In your description, please include how you analyzed the code in the constructor for each of the two classes. Include a description of the instructions that write the vtable pointer into the memory allocated for the instance being constructed (10 points), how you cross checked that what you found was actually the vtable (10 points), and the names/addresses of the functions targeted by the `CustomerFeedback` vtable (5 points).