Calling Convention for Diamondback

1. a)

The caller first stores the arguments required for the function on the stack, then the caller is responsible for moving the return address onto the stack and storing the value of RSP on top of the return address. After that, the caller will move the arguments required for the function to be stored on top of the address of RSP, then the caller will move RSP to point at the return address of the function before calling the function. After the function is called, RSP is still pointing at the return address and we need to set that back. Due to our calling convention, the old value of RSP will always be stored at [rsp-16].

b)

The callee is responsible for getting the arguments on the stack, which are already stored on top of the return addresses, and executing the function and then making sure that the answer to the function is stored in RAX. Furthermore, the callee needs to call ret, which will cause the code to jump back to the return address.

c)

A possible improvement could be to improve the memory usage and efficiency of this calling convention. Due to the setup of the call, we must move the arguments above the return addresses, which means that we have to use double the memory usage to store the same number of arguments. This also means there is inefficiency in having to move the same arguments onto a different part of the stack.

1. a)

b)

c)