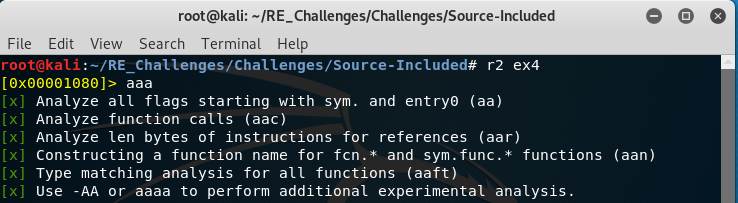
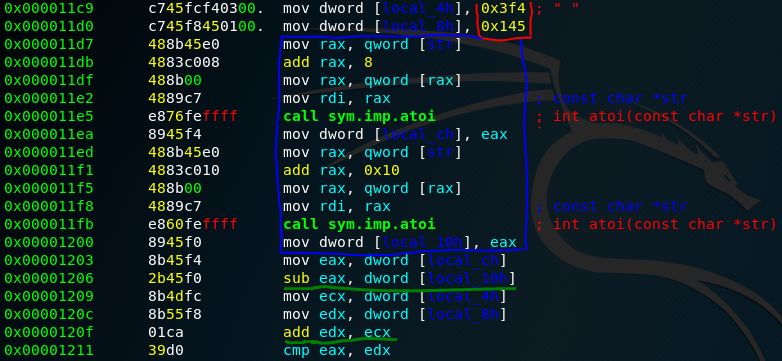
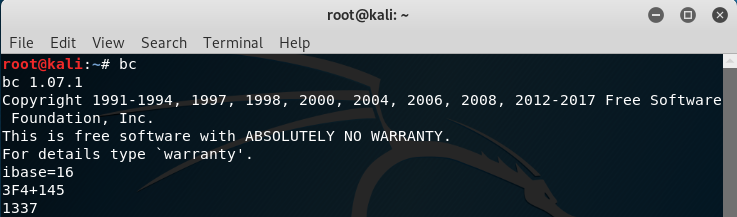
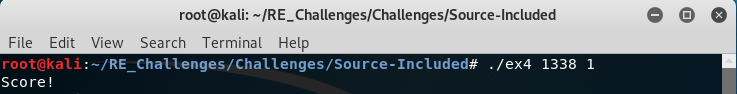
***Ex4***

This exercise takes a new approach, it takes out input strings and converts them to integers and performs mathematical operations on them then compares this number to one that is built during code execution based upon hard coded integers in the binary.

Like before we use radare2 and perform some analysis on it. You are encouraged to follow along using other tools to get a feel for what other tools have to offer. You are also encouraged to play around with the “a” commands to see if things change any, for example just load ex4 into radare2 and try to “[pdf@main](mailto:pdf@main)” and see what happens.

We can see 2 hard coded numbers blocked off in red, a blue block that takes each of our user inputs and converts them to integers with the “atoi” function, ascii to interger, then performs math operations on these numbers underlined in green then compares the results.

Since we know from reading the assembly that the two hard coded values are added together we can copy them and paste them into a calculator to see what value our user inputs will need to evaluate to. For this we will use the simple linux tool “bc”, (basic calculator), once we run the command we will be inside the tools “repl”, (read evaluate print loop), and can start mathing away! However, you will have noticed that these two numbers are in hexadecimal form we will need to change the “ibase”, (input base), variable to base 16 for hexadecimal. The default ibase and obase, output base, variables are set to 10 for decimal numbers so we will not need to change the obase variable for this use case. Now we just have to type in the appropriate mathematical expression, ensuring to capitalize hex letter, and view the result.

Now that we know that the two numbers hard coded into the binary are added together then the result is compared to the subtraction of our two user input values all we have to do is type in two strings that there numerical subtraction results in 1337! Note that this means that there are MANY possibly inputs that can make this program produce a Score!