Phase\_1:

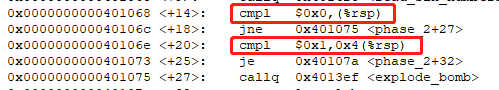
First disas in phase\_1, I see it moves $0x4023a8 to $esi and then it call the function String\_not\_equal which means a compare function. So I need to know what’s in that location.



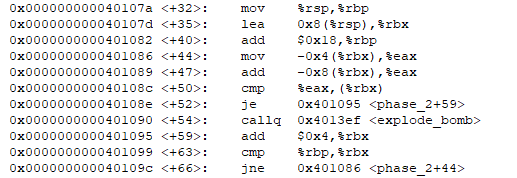
So the string “"He is evil and fits easily into most overhead storage bins." Is the answer.

Phase\_2:

In phase\_2 I see the function <read\_six\_numbers> which means in this phase there will be six numbers as inputs.



In red box, system compares two numbers to %rsp which is stack pointer saved what I inputted. The first number I inputted is saved at the bottom of the stack so it is at %rsp, and the second number is at (%rsp) + 4 and so on. From the picture, it tells that the first number is 0 and the second is 1.



In this picture, there is a loop that compare the third number to the sum of the first and second numbers. So the following numbers are the sum of the two numbers before them. The answer is 0 1 1 2 3 5.

Phase\_3:



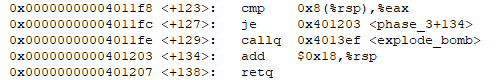
I see it moves 0x4024aa into %esi and then calls a function. I print that location out, the output is "%d %d", which means I need 2 decimal inputs.



Because phase\_3+113 is explode\_bomb, so 0xc(%rsp) has to be less than 0x7



There is a jump and this jump is decided by 0xc(%rsp) which is my first input. I choose 0 here and it will jump to 0x4011bd. %eax is 0x10d here, which is 269 in decimal. Then jump to phase\_3+123.



At this line, it compare the %eax with 0x8(%rsp) which is our second input. So the second input should equals to %eax which is 269.

Two inputs are 0 and 269. This is one solution.

Phase\_4:



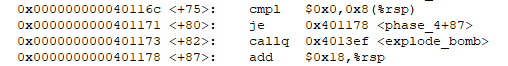
This is very similar to Phase\_3, first I print the location out: "%d %d". The inputs are two decimals.



It first move the first input to %eax and test it. Because <+47> is explode\_bomb, so %eax should greater than zero.



From this picture, I notice that %eax should also less than 14.



From this picture, I notice that my second input should equals to 0.

So I try 1 and 0 as my two inputs, the phase\_4 defused.

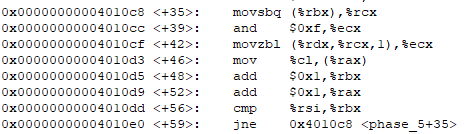
Phase\_5



At the beginning, the program shows me that the input should be a String and the length is 6.



The program moves 0x402430 into %edx, I print it out here: "maduiersnfotvbylInvalid phase%s\n" but I find that %edx is not used in the future, so it doesn’t matter.



This is a loop that make some changes on the value of a letter so that it will become to a new letter.



Here is the target string, I print it out here: "flyers"



In the function Strings\_not\_equal, it is a loop to compare each letter in the input be equal to the letters that saved in %esi. So my mission is to find the letters that after doing the loop will become to “flyers”. The letters are “yonefg”

Phase\_6:

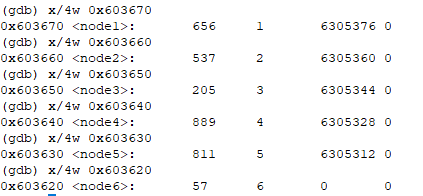
In fact, this is a really long function and I have little confuse on it. So I find some instructions about bomb lab online and try to understand what this function means.



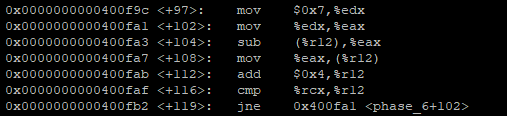
This picture shows that I need 6 number inputs.



At this location, I find out that it has six nodes, that’s are:



My mission is to rearrange this six nodes. The answer is 3 2 6 5 4 1 (7-4, 7-5, 7-1, 7-2, 7-3, 7-6)



Done;