In this assignment, to defuse the bombs I used mainly gdb and given commands in the manual. While using gdb to prevent the bombs to explode I used break for explode_bomb before running and I used in every bomb line again incase of writing the break command as “breake” which resulted in 3 explosions. I also used break in different lines to see to input, output, address, or the value of a register. I also used nexti function time to time for observe the behavior of the code as well as if I were not sure about the code and didn’t want to use next function. Any time I hit a bomb, before exploding it I checked registers by info register method and quit gdb to prevent bomb explosion.

phase_1 :

In phase_1, the code compares the input with a given string. The sting can be found in the address given as #<address> by using x /s <address> method.

phase_2 :

In phase_2, the code first takes 6 inputs. The input must be 6 integers. The rest of the code compares if the given numbers are the first 6 numbers of the Fibonacci series.

phase_3 :

In phase_3, the code needs 2 inputs from the user. The first input must be lower than 7 so I picked 3. Then the code directs a register to a new value which then gets a add up to its address which leads us to stored value. We need to calculate the new address in according to find the second input is compared in a later cmp function.

phase_4 :

In phase_4, the code takes 2 integer inputs. The code first takes my second input and subtract 2 from it and compares if the result is equal bigger than 2. So, for this case I take my second input as 4. Then the code takes my first and second input to func4 where it compares if my first input equals to 12 times of my first input which is 48. Then the code compares if my first input was 48. If it was not 48 the bomb explodes which indicated that 48 4 is the only answer.

phase_5 :

In phase_5, the code takes a string of 6 chars. All the chars indicate a number in an array with 16 indexes. The array is stored at the given address in the code. By printing it I accessed the values of each indexes which goes to 1 to 16 not in an order. The code takes 6 values from the array that have been given by me with the sting input and adds them up and compares the result with 78.

phase_6 :

In phase_6, the code takes 6 integer inputs. Checks if the given numbers are bigger than 0 and lower or equal to 6. Then the code changes all the input values with new values with the given formula
7 – old = new in example 7 – 1 = 6 so it replaces 1 with 6 and does this all the other numbers. Then in the code there is a list and an address for the first node which contains the values for 5 nodes and the address for the next node. From the store of the node 5 I found the address of node 6 than collected all the values for given nodes. Then the code reveals a comparison which checks if the given input and corresponding values are sorted form biggest to lowest.