PHASE 1:
1-I look up the assembly code for phase 1 and saw that there function called strings_not_equal. So I udnerstand that the input should string.
2-I rememberthe pdf file and I run the command "strings bomb"
3-I found a string that looks different than the other strings.
4-I copy that string and run the bomb with that string as my input.

PHASE 2:
1-Firstly I open the assembly code and try to understand what type of input or how many input phase_2 wants.
2-I understand it wants 6 integers when I saw the read_six_numbers method.
3-After reading inputs the code is making a compare the number 1 with the stack pointer
and if it is not 1 the code will explode so I understand that my first input should be 1.
4-Then I saw loop in the code . It was checking if the second input equal to first input plus first input.
5-I saw the pattern and I understand that input should be 1-2-4-8-16-32

PHASE 3:
1-First, I open the assembly code and look inside of the scanf function to the understand what it wants as an inputs.
2-I understand that it wants 2 integers as an input.
3-I put break points to explode_bomb and phase_3 and run my code with two random integers.
4-Then I saw a line after the scanf function and saw that it compares my first input with number 7.
5-I understand that if my first input is greater than number 7 the bomb is going to explode.
6- So I run my code again with my first input as 3 which is less than 7.
7-When my input is less than 7 the code continue to execute and it jumps to an adress. In that adress there was an hexadecimal number and it was puting that number into a register.
8-After that the code compared that integer with my second input and if they are not equal my bomb is going to explode.
9-I run the code with 3 and that integer and it worked.

PHASE 4:
1-First, I open the assembly code and look inside of the scanf function to the understand what it wants as an inputs.
2-I understand that it was wanting 2 integers as an input.
3-After reading the code I understand that it was extracting 2 from one of my input then compare it with 2 and if its not equal to 2 it was going to explode.So I thought that one of my inputs should be 4 or greater.
4-I put my break points to explode_bomb and phase_4 and run my code with 4 and some random integer.But it came to explode function step so I run again this time with random integer and 4 this time.
5-Then I look the info of registers after the execution of the func4 method. There was a number different than inputs in the rax.
6-Then I saw that after it was comparaing eax with the stack pointer (which is equal to my first input in that point) and saw that if they are not equal the bomb is going to explode.

7-So understand that my first input is the return value of the func4. I run my code with that value and 4 and it worked.

PHASE 5:
1-First, I open the assembly code and look inside of the code I could not see a scanf function this time but I saw the string_length function and I assume that it wants a string as an input.
2-After the string_length function there was a compare with 6 and return value. When the return value is not 6 the bomb was exploding.So I assume that the length of the string should be 6.
3-I put my break points to explode_bomb and phase_5 and run my code with some random string with 6 chars.
4-After few line there was an hashtag and array and its adress.So I look up to address of that array.Inside of the array there was 16 random chars and a sentence from dr evil which I assume the sentece does not matter.
5-Then I saw an another hashtag and address so I look up to that adress and I found there was an another string which was "bruins".
6-I look up the code between these two addresses and I try to understand what is the relationship and what is the code doing.
7-After looking inside of the registers to understand the relationship between my input and the other strings I realized that my input characters are corresponds to their alphabetic order like a is 1 b is 2 c is 3 ...
8-When I try to understand the comparion operations I noticed that the characters in the "bruins" with my input string.Also "bruins" string was sorted by the 16 characters in the first string, according to which element they are in that array. bruins was correponds to 13 6 3 4 8 7 . So I choose chars that corresponds to these numbers in the alphabet. It was mfcdhg.I try this as my input and it worked.

PHASE 6:
1-First, I open the assembly code and look inside of the code I could not see a scanf function but I saw the read_six_numbers function and I assume that it wants 6 integers as an input.
2-I put my break points to explode_bomb and phase_6 and run my code with some random 6 integers.
3-Then I saw that first it was adding 1 to a register then it was comparing this register with 5.At that point that register was containing my first input number.It was like a loop. So assume that my inputs should be lower or equal to 6.
4-So I give my input as 1 2 3 4 5 6 and run it again.
5-Then when I look at the assembly code there was a lea command and a address near the command. I look up the address and in the end I found six nodes in that adress. All of them were containing a hexadecimal number and an adress. The adresses were equal to the adress of the next node. I write all of these to a piece of paper.
6-After that I saw a lot of mov instructions and comparison instruction at the end. I go to comparison line and look up to info of registers.
7-It was a comparison of the value of the nodes with value of the next node.Since there is 6 input and 6 node and since the inputs can not be larger than 6 I though that I should sort the inputs according to order of the number values of the nodes from small to large.It did not work at first try then I sort them to large to small.It worked this time.