Q1: What is the current value of i? What statement in the code causes the value of i to be what it is? Explain why the statement causes i to be what it is.
Answer: The current value of i equals to 1368. The statement in the code that causes the value of i to be is sum + = a[i]. Sum + =a[i] means sum = sum + array[i], so it sums up the value of array[i] to the variable sum. In the file he gives it says 1<size instead of i<size which makes the for loop keep going because 1<size is always true. Once it comes up to 1368 the value after that is out of the memory that is provided which gives the segmentation fault. The value of i is the type of variable that declares the sum of the array. sum += a[i] doesn't change the value of i, only the value of sum.

Question 2*. Repeat command* **s** *30 times. What are the names of the functions that are called when you do this?*

```
Breakpoint 1, power (base=3, exponent=4) at lab3support.c:54
warning: Source file is more recent than executable.
54                  return base * power(base , exponent -1);
(gdb) s
power (base=3, exponent=3) at lab3support.c:49
49            if(exponent ==0)
(gdb) s
51            else if(exponent == 1)
(gdb) s
```

Question 3*. What statement in the code causes the error in the program? Explain why the statement causes the error. Justify your answer by information that you got when running the debugger.*
Answer: The statement that had the error was in the else statement. The error in the program is when we run the program the program doesn't change the value of base or exponent. When running the debugger, the else statement is not implemented to change values and keeps returning the value 3 for base and 4 for exponent each time the command s is used.

Question 4*. How many function calls does the program make before the recursive function starts returning? (****Count main as the first call****. Hint: you can break at the last recursive call then use where command to see the stack frames for each function call).*

Answer: there are going to be five function calls until the program recursive function. As it is 4 for the recursive method and 1 to call to the function.