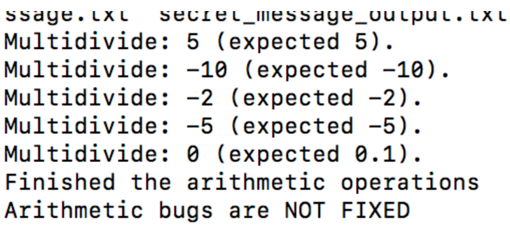
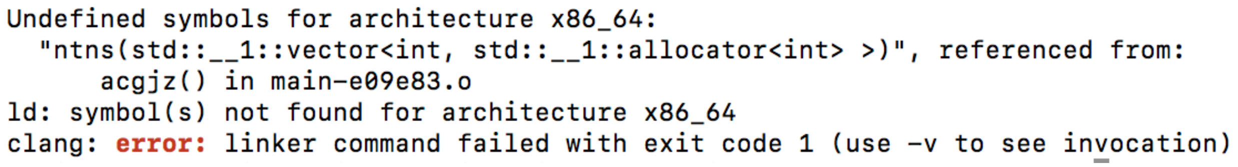
My operating system is IOS system, because my laptop is MAC book. My compiler is my computer terminal and I used the X code to write my code, which I download from the apple app store.  At first, I start from the simplest one the arithmetic test. After I first ran it, I can see from the terminal output that I did not get the expected value, thus from that on I understood one of the important missions for me to fix all the bugs in which to get the expected and pass all the correct "assert" statement.  From the line number, I can know where I was wrong, then I began my calculation first to get the expected value which listed in each line. All thing went well. However, when I get into the final expected output, which is a float number, I realized one knowledge point --- the integer divided the integer only can get the integer. Therefore, I tried to add the float to the last line of "getting the number codes." I stilled failed. I did the recalculation, but I cannot find anything wrong with my calculations. The only way for me is to focus something outside the main arithmetic operation function but still used in the main arithmetic operation function, which is my "brko" function for calculating the long division operation. I got the shock the dead problem is this small function outside where I focus most --- I missed here is still needed to add "float" in order to the decimal calculation answer.



```
ssage.txt  secret_message_output.txt
Multidivide: 5 (expected 5).
Multidivide: -10 (expected -10).
Multidivide: -2 (expected -2).
Multidivide: -5 (expected -5).
Multidivide: 0 (expected 0.1).
Finished the arithmetic operations
Arithmetic bugs are NOT FIXED
```

I started my "file" common line debugging.  I just first ran this part, there is nothing output, which leading me begin to look for the "cerr" and 'cout" statement to find out why there is nothing output there even there is no compile error or "segmentation fault." When I compared my compile command line in my terminal, I noticed that the number of arguments and fix this error of "equal" or "not equal" the number of command line. I read through the codes and similarly fix the error of "right" and "not right" error. Coming to the final bug, I fixed in the "file-read" operation part, I used the "std::cout" to check the total bytes of the data, and I find there is no errors of the "read" function and decided to take some changes on the "assert" one. And finally, I went through this one.

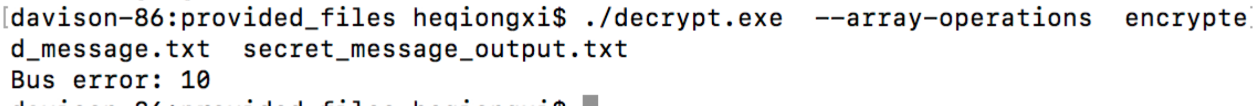
As I first ran the "vector" operation, I get the terminal error information – "Segmentation fault: 11", pointing me to the trivial function in the "assert" one. In the "ntns" function, I saw the bug of the index equal the size of the vector, so I change the index from "1" instead from "0" and take out the "=" symbol.  After correcting this one, I faced one error which is confused for me, which informed me that "linker command failed with exit code 1 (use -v to see invocation)." I though "list" always has something to do with link. Is something wrong with my address? I began to seek my declarations of functions, and I notice that I miss the "&" the address symbol in the top declarations' parts. Then, I remember the most impressive part for me is the "for loop" in the middle part of the main vector operating function.  When I get the "cout" outputs, the index of my output is very strange like some number up to "-16804". In this way, I know there must some errors in the for loop before "cout" statement. The number counting of the value "is divisible by 3" catches my attention.  I began use "std::cout" to get all the process of counting the value, and I finally find that just "int variable" is not unworkable, so I gave it default value zero. What’s more, looking through the loop, this loop did not put its value in the vector, just the index. The right way to do this one is to access the value by that index. When I get this point, although all outputs are correct, there is an output in terminal "Vector bugs are NOT FIXED". I look back that loop this error intrigue me interest: unsigned integer cannot be negative, however I included the negative one in “unsigned int” range, which the terminal report “segmentation fault.” This taught me that the range of for loop is very important.



```
Undefined symbols for architecture x86_64:
  "ntns(std::__1::vector<int, std::__1::allocator<int> >)", referenced from:
      acgjz() in main-e09e83.o
ld: symbol(s) not found for architecture x86_64
clang: error: linker command failed with exit code 1 (use -v to see invocation)
```

Talking about the list operation, the most unforgotten bug for me is the distinguish between the address and the value in that address. The logic of that part of code asks me to remove the number satisfying the request of multiple of given numbers. At first, I didn’t find anything wrong with the logic of codes in here, but I didn’t get the expected output --- thus I think something wrong with the “for loop” because of I didn’t get the right answers when I add “std::cout” on here. When I look close to the iterator value of the for loop, I find that the value in the “if” statement is just pointer not the value in that pointer. Moreover, going to the next bugs, when I wonder the why the “for” loop break, I realize the “for” loop would break all the “for” loop, so I changed it into “continue” for the keeping doing on the loop.

When I first ran the “’array” operation, there is the “segmentation fault” and nothing printed out. I consider the building “2D array” codes are not successful. It is the common problem of “out of bound.” The most interesting bug for is the logic function bug --- the function of getting the “hypotenuse” one. I tested multiple examples, thus I go the conclusion of when the first parameter is less than the second one, it would get the negative hypotenuse which is impossible. Initially, I wanted to add some if loops, but I thought it was too complicated. Therefore, I was considering the way can let me to change the number to positive one, I chose to add “abs’ math function.



```
[davison-86:provided_files heqiongxi$ ./decrypt.exe  --array-operations  encrypte
d_message.txt  secret_message_output.txt
Bus error: 10
```

Above, I debug this file pretty much by my reading and understanding of these codes, and by using “std:;cout’ help me to find out the differences between my output and expected output. Moreover, the terminal bug statement also helps me a lot.