Environment Info:

**Systems / Hardware**

1. **macOS Mojave**

**Version** 10.14.2 (18C54)

**MacBook Pro** (13-inch, 2017, Four Thunderbolt Ports)

**Processor** 3.1 GHz Intel Core i5

**Memory** 8 GB 2133 MHz LPDDR3

**Graphics** Intel Iris Plus Graphics 650 1536 MB

1. **Ubuntu 18.04.1 LTS**

**Memory** 31.4 GiB

**Processor** AMD Ryzen 7 1700 eight-core processor x 16

**Graphics** GeForce GTX 1080/PCIe/SSE2

**GNOME** 3.28.2

**OS type** 64-bit

**Disk** 117.6 GB

**Compliers**

1. **macOS: Apple LLVM version 10.0.0 (clang-1000.11.45.5)**

Configured with: --prefix=/Applications/Xcode.app/Contents/Developer/usr --with-gxx-include-dir=/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.14.sdk/usr/include/c++/4.2.1

Apple LLVM version 10.0.0 (clang-1000.11.45.5)

Target: x86\_64-apple-darwin18.2.0

Thread model: posix

InstalledDir: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin

1. **Ubuntu 18.04.1 LTS**

g++ (Ubuntu 7.3.0-27ubuntu1~18.04) 7.3.0

Copyright (C) 2017 Free Software Foundation, Inc.

This is free software; see the source for copying conditions.  There is NO

warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**Text Editor/ IDE**

1. **Sublime**

3.1.1 (BUILD 3176) 14 May 2018

1. **CLion 2018.3.4**

Build #CL-183.5429.37, built on January 31, 2019

Licensed to Zhicheng Guo

Subscription is active until August 30, 2019

For educational use only.

JRE: 1.8.0\_152-release-1343-b26 x86\_64

JVM: OpenJDK 64-Bit Server VM by JetBrains s.r.o

macOS 10.14.2

**Memory Debugger:**

Linux 2.0.1 Release of Dr. Memory

Report Structure:

Each bug case will be presented in the following form

**ERROR #number:**

**File:** filename

**Arguments:** arguements

**Output:** some error or abnormal output

**Original Code:**

foo.cpp:#line number

CODEBLOCKS

**Debug Process:**

1. BLABLABA
2. BLABLABALBLABAL

**Debugged Code** (changes are indicated by red background):

DEBUGGED CODEBLOCKS

**Memory ERROR #number**

**File:** filename

**Arguments:** arguements

**Memory Erros:** dr memory output, screenshots

**Original Code:**

foo.cpp:#line number

CODEBLOCKS

**Debug Process:**

1. BLABLABA
2. BLABLABALBLABAL

**Debugged Code** (changes are indicated by red background):

DEBUGGED CODEBLOCKS

Bug fixing:

Arithmetic Operations:

**ERROR 1:**

**File: main.cpp**

**Arguments:** --arithmetic-operations encrypted\_message.txt secret\_message\_output.txt

**Output:** Assertion failed: (brko(ancy,einv\_,ixgn,5,einv\_) == 5), function dambeq, file /Users/cheng\_stark/Desktop/DS-2019/HW4\_Clion/main.cpp, line 336.

**Original Code:**

main.cpp: 336

**assert**(brko(ancy,einv\_,ixgn,5,einv\_) == 5);

main.cpp: 304

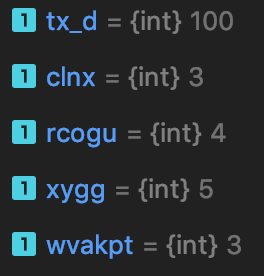
float brko(int tx\_d, int clnx, int rcogu, int xygg, int wvakpt) {  
 float wzhuae = ((((tx\_d / clnx) / rcogu) / xygg) / wvakpt);  
 return wzhuae;  
}

**Debug Process:**

1. First glance, there is a type mismatch between int and float. The dividing process will be casted into int and gives out exact. So I change it to float to get the real division .
2. The error still exist. Then I break at line 336

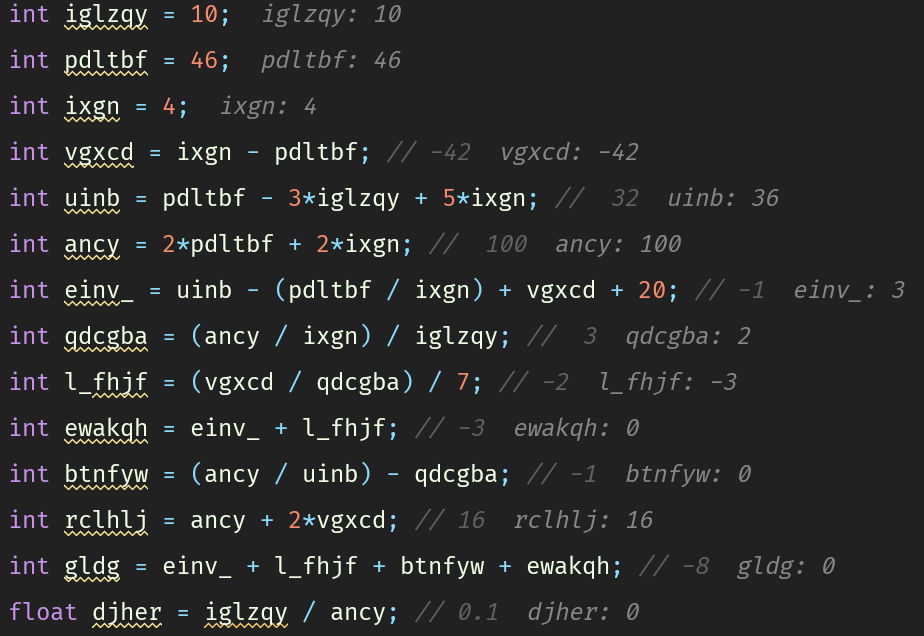


And then I step in



However, these values shoule be

1. I went up and found out that there are math erros, since resultls are wrong(indicated by blue underlines), I did some calculations and changed the code.



**Debugged Code** (changes are indicated by red background):

*// set up some variables*int iglzqy = 10;  
int pdltbf = 46;  
int ixgn = 4;  
int vgxcd = ixgn - pdltbf; *// -42*int uinb = pdltbf - 3\*iglzqy + 4\*ixgn; *// 32*int ancy = 2\*pdltbf + 2\*ixgn; *// 100*int einv\_ = uinb - (pdltbf / ixgn) + vgxcd + 20; *// -1*int qdcgba = (ancy / ixgn) / iglzqy + 1; *// 3*int l\_fhjf = (vgxcd / qdcgba) / 7; *// -2*int ewakqh = einv\_ + l\_fhjf; *// -3*int btnfyw = (ancy / uinb) - qdcgba - 1; *// -1*int rclhlj = ancy + 2\*vgxcd; *// 16*int gldg = einv\_ + l\_fhjf + btnfyw + ewakqh - 1; *// -8*float djher = iglzqy / ancy; *// 0.1*

File Operations:

**ERROR 2:**

**File: main.cpp**

**Arguments:** --file-operations encrypted\_message.txt secret\_message\_output.txt

**Output:** Usage: /Users/cheng\_stark/Desktop/DS-2019/HW4\_Clion/cmake-build-debug/HW4\_Clion operations infile outfile

Couldn't start operations.

**Original Code:**

main.cpp:252

*// Error checking on command line arguments*if(argc == 4) { // logical bug here  
 std::cerr << "Usage: " << argv[0] << " operations infile outfile" << std::endl;  
 std::cerr << "Couldn't start operations." << std::endl;  
 return false;  
}

**Debug Process:**

1. After the output, according to the error message found the code segment.
2. The error hould only be raised if argv is not 4

**Debugged Code** (changes are indicated by red background):

if(argc != 4) {  
 std::cerr << "Usage: " << argv[0] << " operations infile outfile" << std::endl;  
 std::cerr << "Couldn't start operations." << std::endl;  
 return false;  
}

**ERROR 3:**

**File: main.cpp**

**Arguments:** --file-operations encrypted\_message.txt secret\_message\_output.txt

**Output:**

That file could not be opened!

**Original Code:**

main.cpp:265

if(ypu\_) {

**Debug Process:**

1. Obviouse logical error, should raise error when the file is invalid not valid.

**Debugged Code** (changes are indicated by red background):

if(!ypu\_) {

**ERROR 4:**

**File: main.cpp**

**Arguments:** --file-operations encrypted\_message.txt secret\_message\_output.txt

**Output:** Assertion failed: (ypu\_.gcount() != wyom), function uus\_yg, file /Users/cheng\_stark/Desktop/DS-2019/HW4\_Clion/main.cpp, line 289.

**Original Code:**

main.cpp:289

**assert**(ypu\_.gcount() != wyom);

**Debug Process:**

1. Obviouse logical error, should raise error when the file length and array length are not equal.

**Debugged Code** (changes are indicated by red background):

**assert**(ypu\_.gcount() == wyom);

Array Operations:

**ERROR 5:**

**File: main.cpp**

**Arguments:** --array-operations encrypted\_message.txt secret\_message\_output.txt

**Output:** Process finished with exit code 10

**Original Code:**

main.cpp:407

*// Error checking on command line arguments*for(int gwcnpy=1; gwcnpy<=hmgo; ++gwcnpy) {  
 ljdsq[gwcnpy] = new int[hmgo];  
 bdfbc[gwcnpy] = new int[hmgo+1];  
 for(int wvp\_u=1; wvp\_u<=hmgo; ++wvp\_u) {  
 ljdsq[gwcnpy][wvp\_u] = 0;  
 ljdsq[gwcnpy+1][wvp\_u+1] = 0; //error here  
 }  
}

*//sanity check: corners of array*

**assert**(ljdsq[1][1] == 0);  
**assert**(ljdsq[1][-1] == 0);  
**assert**(ljdsq[-1][1] == 0);  
**assert**(ljdsq[-1][-1] == 0);

**Debug Process:**

1. The code 10 means the memory is not initialized
2. The starting index for ljdsq is obviously incorrect, should have started with 0 other wise the 0th slot will not be initialized
3. [-1] is not a valid operation as it points a slot previouse to the start of the array

**Debugged Code** (changes are indicated by red background):

for(int gwcnpy=1; gwcnpy<=hmgo; ++gwcnpy) {  
 ljdsq[gwcnpy-1] = new int[hmgo];  
 bdfbc[gwcnpy] = new int[hmgo+1];  
 for(int wvp\_u=1; wvp\_u<=hmgo; ++wvp\_u) {  
 std::cout<<gwcnpy-1<<"||"<<wvp\_u-1<<std::endl;  
 ljdsq[gwcnpy-1][wvp\_u-1] = 0;  
 bdfbc[gwcnpy][wvp\_u] = 0;  
 }  
}  
  
*// sanity check: corners of array***assert**(ljdsq[1][1] == 0);  
**assert**(ljdsq[1][24] == 0);  
**assert**(ljdsq[24][1] == 0);  
**assert**(ljdsq[24][24] == 0);

**ERROR 6:**

**File: main.cpp**

**Arguments:** --array-operations encrypted\_message.txt secret\_message\_output.txt

**Output:** Assertion failed: (ljdsq[1][2] == -1), function nw\_jtj, file /Users/cheng\_stark/Desktop/DS-2019/HW4\_Clion/main.cpp, line 432.

**Original Code:**

main.cpp:432

**assert**(ljdsq[1][2] == -1); *// no triple exists*

main.cpp:420

for(int gwcnpy=1; gwcnpy>=hmgo; ++gwcnpy) {  
 for(int wvp\_u=1; wvp\_u>=hmgo; ++wvp\_u) {  
 ljdsq[gwcnpy][wvp\_u] = panjm(gwcnpy, wvp\_u);  
 bdfbc[gwcnpy][wvp\_u] = ljdsq[gwcnpy][wvp\_u] \* panjm(wvp\_u, gwcnpy);  
 }  
}

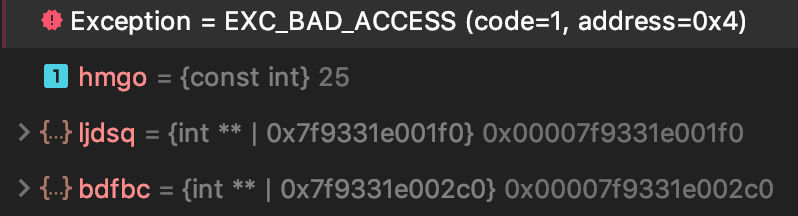
main.cpp:95

int panjm(int jhkg, int ozns) {  
 double\* wvdst; *// will store the integer part from modf  
 // read up on modf with "man modf" in your terminal  
  
 // jhkg and y are both legs* float tnxq = jhkg\*jhkg + ozns\*ozns;  
 float rspic = modf(sqrt(tnxq), wvdst);  
 if((rspic = 0))  
 return (int) \*wvdst;  
  
 *// jhkg is the hypotenuse, need to subtract instead of add* float aiwwyc = ozns\*ozns - jhkg\*jhkg;  
 rspic = modf(sqrt(aiwwyc), wvdst);  
 if((rspic = 0))  
 return (int) \*wvdst;  
  
}

**Debug Process:**

1. The debugging mode of Clion show BAD\_ACCESS

BAD\_ACCESS usually indicates that there is problem with the pointer



1. An obviouse type mismatch between float and double
2. Also since types are all float, we should use modff instead of modf, since modf is for double (I googled the modf operation, from https://en.cppreference.com/w/c/numeric/math/modf)

|  |
| --- |
| float       modff( float arg, float\* iptr ); |
| double      modf( double arg, double\* iptr ); |
| long double modfl( long double arg, long double\* iptr ); |  |  |

1. Since this is a int function, the function now reaches the end but no return. It should return something at the end if there is no triple.
2. Logical sign = should be ==
3. The differenc between ozns\*ozns - jhkg\*jhkg should be positive at all time
4. \* should not be used here, the wvdst should only be on stack
5. This didn’t resolve the issue, then I checked other codes
6. There is an error in for loop conditions the >= should be <

**Debugged Code** (changes are indicated by red background):

for(int gwcnpy=1; gwcnpy<hmgo; ++gwcnpy) {  
 for(int wvp\_u=1; wvp\_u<hmgo; ++wvp\_u) {  
 ljdsq[gwcnpy][wvp\_u] = panjm(gwcnpy, wvp\_u);  
 bdfbc[gwcnpy][wvp\_u] = ljdsq[gwcnpy][wvp\_u] \* panjm(wvp\_u, gwcnpy);  
 }  
}

int panjm(int jhkg, int ozns) {  
 float wvdst; *// will store the integer part from modf  
 // read up on modf with "man modf" in your terminal  
  
 // jhkg and y are both legs* float tnxq = jhkg\*jhkg + ozns\*ozns;  
 float rspic = modff(sqrt(tnxq), &wvdst);  
 if(rspic == 0)  
 return (int) wvdst;  
  
 *// jhkg is the hypotenuse, need to subtract instead of add* float aiwwyc = abs(ozns\*ozns - jhkg\*jhkg);  
 rspic = modff(sqrt(aiwwyc), &wvdst);  
 if(rspic == 0)  
 return (int) wvdst;  
  
 return -1;  
}

**ERROR 7:**

**File: main.cpp**

**Arguments:** --array-operations encrypted\_message.txt secret\_message\_output.txt

**Output:** Exception: EXC\_BAD\_ACCESS (code=EXC\_I386\_GPFLT)

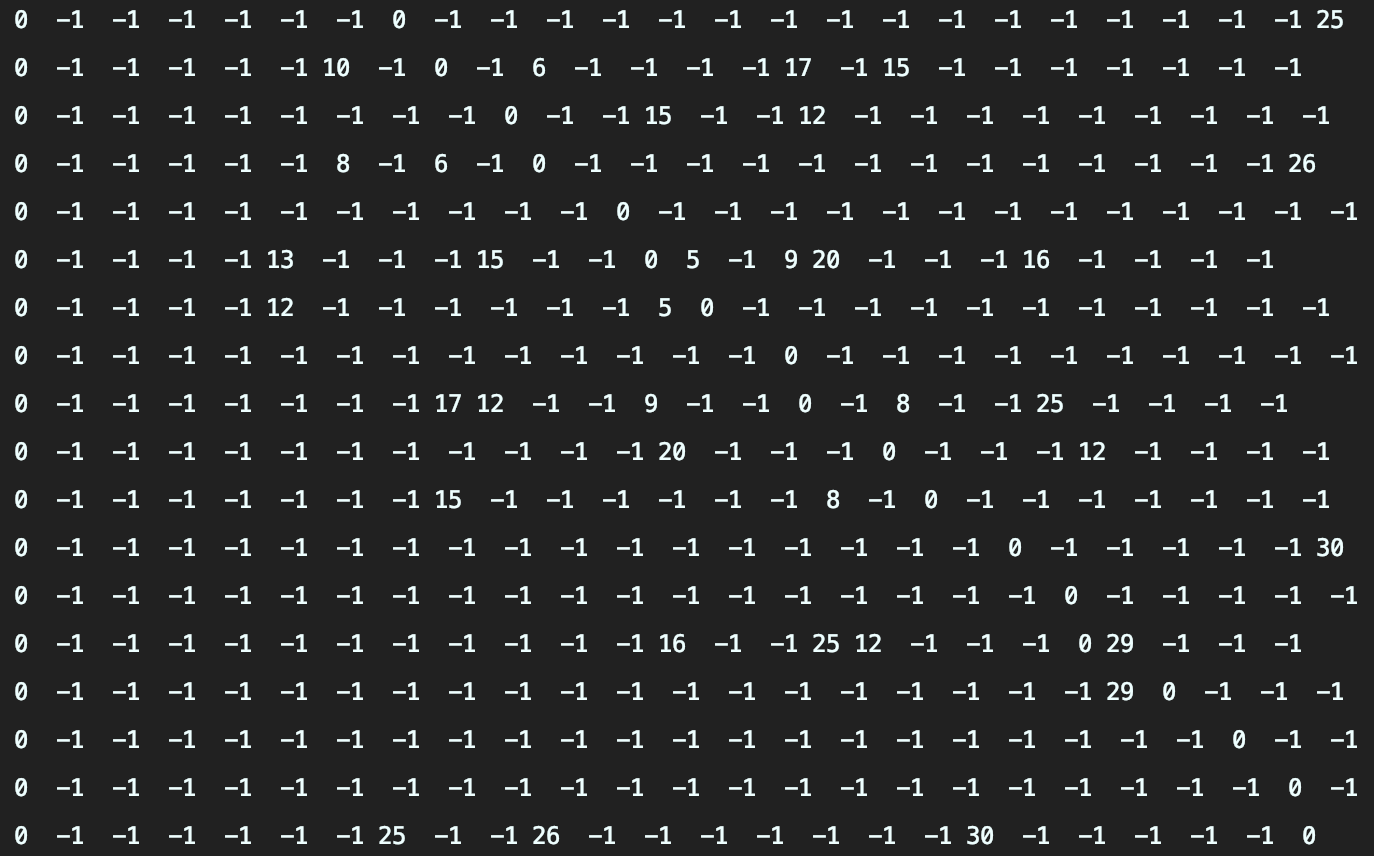
**Original Code:**

main.cpp:448

int\*\* fawbg = ljdsq;  
int\*\* aadvzu = bdfbc;  
for(int gwcnpy = 1; gwcnpy <= hmgo; ++gwcnpy, ++fawbg, ++aadvzu) {  
 int\* kvctc = \*fawbg;  
 for(int wvp\_u = 1; wvp\_u <= hmgo; ++wvp\_u, ++fawbg) {  
 int vhnuig = \*kvctc;  
 *// pad single-digit numbers with a space so it looks nice  
 // ain't nobody got time for <iomanip>* std::string edrrn = ((vhnuig < 10) ? " " : "");  
 std::cout << edrrn << \*kvctc << " ";  
 }  
 std:: cout << std::endl;  
}

**Debug Process:**

1. <= is a very obviouse for loop condition error
2. The starting index should all be 0
3. In line for(int wvp\_u = 1; wvp\_u <= hmgo; ++wvp\_u, ++fawbg) {, it is just increment fawbg while it should increment kvctc
4. Output are not formatted after testing, so I searched online for the usage of ?(question mark) and added && vhnuig >= 0



**Debugged Code** (changes are indicated by red background):

int\*\* fawbg = ljdsq;  
int\*\* aadvzu = bdfbc;  
for(int gwcnpy = 0; gwcnpy < hmgo; ++gwcnpy, ++fawbg, ++aadvzu) {  
 int\* kvctc = \*fawbg;  
 for(int wvp\_u = 0; wvp\_u < hmgo; ++wvp\_u, ++kvctc) {  
 int vhnuig = \*kvctc;  
 *// pad single-digit numbers with a space so it looks nice  
 // ain't nobody got time for <iomanip>* std::string edrrn = ((vhnuig < 10&& vhnuig >= 0) ? " " : "");  
 std::cout << edrrn << \*kvctc << " ";  
 }  
 std:: cout << std::endl;  
}

Vector Operations:

**ERROR 8:**

**File: main.cpp**

**Arguments:** --vector-operations encrypted\_input.txt secret\_message\_output.txt

**Output:** EXC\_BAD\_ACCESS (code=1, address=0x7fcf1d40237c)

**Original Code:**

main.cpp:29

int ntns(std::vector<int> eujjll);

main.cpp:763

int ntns(std::vector<int> eujjll) {  
 for(uint vfpdky=0; vfpdky<=eujjll.size(); ++vfpdky) {  
 eujjll[vfpdky] = eujjll[vfpdky] + eujjll[vfpdky-1];  
 }  
 return eujjll[eujjll.size()];  
}

**Debug Process:**

1. We do want to change this vector and preserve the changes we made out side this function, so we need to add a &
2. For loop indexing have some problems, <= will go out of bound and index should start from 1
3. Since we started from index 1, the total length would be one less

**Debugged Code** (changes are indicated by red background):

int ntns(std::vector<int> &eujjll);

int ntns(std::vector<int> &eujjll) {  
 for(uint vfpdky=1; vfpdky<eujjll.size(); ++vfpdky) {  
 eujjll[vfpdky] = eujjll[vfpdky] + eujjll[vfpdky-1];  
 }  
 return eujjll[eujjll.size()-1];  
}

**ERROR 9:**

**File: main.cpp**

**Arguments:** --vector-operations encrypted\_input.txt secret\_message\_output.txt

**Output:** Assertion failed: (gegxun == 55), function acgjz, file /Users/cheng\_stark/Desktop/DS-2019/HW4\_Clion/main.cpp, line 533.

**Original Code:**

main.cpp:533

**assert**(gegxun == 55);

main.cpp:517

for(uint hxzpt=0; hxzpt<10; ++hxzpt) {  
 lu\_bu.push\_back(hxzpt);  
}

**Debug Process:**

1. According to comment, it should be 1-10 while actually it is now 0-9

**Debugged Code** (changes are indicated by red background):

for(uint hxzpt=1; hxzpt<=10; ++hxzpt) {  
 lu\_bu.push\_back(hxzpt);  
}

**ERROR 10:**

**File: main.cpp**

**Arguments:** --vector-operations encrypted\_input.txt secret\_message\_output.txt

**Output:** Assertion failed: (lkfe == 4), function acgjz, file /Users/cheng\_stark/Desktop/DS-2019/HW4\_Clion/main.cpp, line 553.

**Original Code:**

main.cpp:553

**assert**(lkfe == 4);

main.cpp:545

int lkfe;

**Debug Process:**

1. The varible is not initialized, which should be initialized to 0

**Debugged Code** (changes are indicated by red background):

int lkfe = 0;

**ERROR 11:**

**File: main.cpp**

**Arguments:** --vector-operations encrypted\_input.txt secret\_message\_output.txt

**Output:** Assertion failed: (gersa(toxx\_, mwwpvp)), function acgjz, file /Users/cheng\_stark/Desktop/DS-2019/HW4\_Clion/main.cpp, line 580.

**Original Code:**

main.cpp:580

**assert**(gersa(toxx\_, mwwpvp));

main.cpp:479

bool gersa(const std::vector<int> togp, const std::vector<int> twobsk) {  
 bool tgzbpv = true;  
 for(uint gfprfg=0; gfprfg<togp.size(); ++gfprfg) {  
 if(togp[gfprfg] > twobsk[gfprfg]) {  
 tgzbpv = false;  
 }  
 }  
 return tgzbpv;  
}

**Debug Process:**

1. Logical error with the > sign, it should be <= according to the comment
2. The loop should break after one false instance

**Debugged Code** (changes are indicated by red background):

bool gersa(const std::vector<int> togp, const std::vector<int> twobsk) {  
 bool tgzbpv = true;  
 for(uint gfprfg=0; gfprfg<togp.size(); ++gfprfg) {  
 if(togp[gfprfg] <= twobsk[gfprfg]) {  
 tgzbpv = false;  
 break;  
 }  
 }  
 return tgzbpv;  
}

**ERROR 12:**

**File: main.cpp**

**Arguments:** --vector-operations encrypted\_input.txt secret\_message\_output.txt

**Output:** Now counting numbers divisible by 3 Process finished with exit code 9

**Original Code:**

main.cpp:602

std::cout << "Now counting numbers divisible by 3" << std::endl;  
for(uint ujie = 0; ujie < njtt.size(); ujie+1) {  
 if(ujie % 3 == 0) {  
 *// std::cout << njtt[ujie] << " is divisible by 3" << std::endl;* lkfe++;  
 mxbgxq.push\_back(ujie);  
 }  
}  
std::cout << "There are " << lkfe << " numbers divisible by 3."  
 << std::endl;

*// loop over it and print it out backwards*for(uint vsqn=lkfe-1; vsqn >= 0; --vsqn) {  
 std::cout << "mxbgxq[" << vsqn << "] = " << mxbgxq[vsqn] << std::endl;  
}

**Debug Process:**

1. There is no lkfe, we should initialize it to 0
2. We should use njtt[ujie] to divide instead of ujie (just the index is wrong, we need the item of the index)
3. We should store the njtt[ujie] for the same reason as above
4. Unit is unsigned int which is >=0 while -1 is not an unsigned int, we should use int instead
5. Obviouse erro ujie+1 should be ujie++

**Debugged Code** (changes are indicated by red background):

std::cout << "Now counting numbers divisible by 3" << std::endl;  
lkfe = 0;  
for(uint ujie = 0; ujie < njtt.size(); ujie++) {  
 if(njtt[ujie] % 3 == 0) {  
 *// std::cout << njtt[ujie] << " is divisible by 3" << std::endl;* lkfe++;  
 mxbgxq.push\_back(njtt[ujie]);  
 }  
}  
std::cout << "There are " << lkfe << " numbers divisible by 3."  
 << std::endl;  
  
*// loop over it and print it out backwards*for(int vsqn=lkfe-1; vsqn >= 0; --vsqn) {  
 std::cout << "mxbgxq[" << vsqn << "] = " << mxbgxq[vsqn] << std::endl;  
}

List Operations:

**ERROR 13:**

**File: main.cpp**

**Arguments:** --list-operations encrypted\_input.txt secret\_message\_output.txt

**Output:** Assertion failed: (idjwy.back() == 'z'), function wlid, file /Users/cheng\_stark/Desktop/DS-2019/HW4\_Clion/main.cpp, line 124.

**Original Code:**

main.cpp:124

**assert**(idjwy.back() == 'z');

main.cpp:117

std::list<char> idjwy;  
for(char lggvlu = 'a'; lggvlu >= 'z'; lggvlu++) {  
 idjwy.push\_back(lggvlu);  
 }  
 for(char lggvlu = 'A'; lggvlu <= 'Z'; lggvlu++) {  
 idjwy.push\_front(lggvlu);  
 }

**Debug Process:**

1. A logic error, Since it is a push\_front operation, we should reverse from Z to A instead
2. Alsoa a logic error, it should go from a to z, it should be <= ‘z’

**Debugged Code** (changes are indicated by red background):

std::list<char> idjwy;  
for(char lggvlu = 'a'; lggvlu <= 'z'; lggvlu++) {  
 idjwy.push\_back(lggvlu);  
 }  
 for(char lggvlu = 'Z'; lggvlu >= 'A'; lggvlu--) {  
 idjwy.push\_front(lggvlu);  
 }

**ERROR 14:**

**File: main.cpp**

**Arguments:** --list-operations encrypted\_input.txt secret\_message\_output.txt

**Output:** elderberry quart nectarine orange zwetschge pomegranate durian grape yellow squash fig iodine strawberry tangerine jujube lemon mango cherry uglyfruit apple watermelon kiwi

256494680 letters did not ever appear in the fruit names.

List bugs are NOT FIXED

**Original Code:**

main.cpp:138

for(std::list<int>::iterator btimc = ckgdyc.begin(); btimc != ckgdyc.end(); ++btimc) {  
 if(\*btimc % tbox != 0 || \*btimc % ncnp != 0) {  
 ckgdyc.erase(btimc);  
 }  
}

main.cpp:206

int wdper;  
for(std::list<char>::iterator wjwci = idjwy.end(); wjwci != idjwy.begin(); wjwci--) {  
 if(\*wjwci < 'a' || \*wjwci > 'z') {  
 break;  
 }  
 wdper++;  
}

main.cpp:182

std::list<std::string>::iterator yfxeo;  
for(std::list<std::string>::reverse\_iterator qjdob = qangui.rbegin();  
 qjdob != qangui.rend(); qjdob++) {  
 yfxeo = std::find(flbfdw.begin(), flbfdw.end(), \*qjdob);  
 flbfdw.erase(++yfxeo);  
}

**Debug Process:**

1. A logic error, according to comment, only things that are *remove every number from the list that is a multiple of at least one of these ssmxiz,* so != should be == instead
2. Also when I erase something from the list, the iterator should not be incrementing
3. In flbfdw.erase(++yfxeo); the iterator yfxeo has no point of incrementing
4. wdper should be initialized to 0
5. After I made all fixes above, I still get output of ” 0 letters did not ever appear in the fruit names.” Then I tried to cout each letter, I found out idjwy.end() does not have a specific meaning in the list and can not access any values via \*, so change it to –idjwy.end() instead.

**Debugged Code** (changes are indicated by red background):

for(std::list<int>::iterator btimc = ckgdyc.begin(); btimc != ckgdyc.end(); ++btimc) {  
 if(\*btimc % tbox == 0 || \*btimc % ncnp == 0) {  
 ckgdyc.erase(btimc--);  
 }  
}

int wdper = 0;  
for(std::list<char>::iterator wjwci = --idjwy.end(); wjwci != idjwy.begin(); wjwci--) {  
 if(\*wjwci < 'a' || \*wjwci > 'z') {  
 break;  
 }  
 wdper++;  
}

std::list<std::string>::iterator yfxeo;  
for(std::list<std::string>::reverse\_iterator qjdob = qangui.rbegin();  
 qjdob != qangui.rend(); qjdob++) {  
 yfxeo = std::find(flbfdw.begin(), flbfdw.end(), \*qjdob);  
 flbfdw.erase(yfxeo);  
}

All Operations:

**ERROR 15:**

**File: main.cpp**

**Arguments:** --all-operations encrypted\_input.txt secret\_message\_output.txt

**Output:** Assertion failed: (igbm(jd\_vg, djher)), function dambeq, file /Users/cheng\_stark/Desktop/DS-2019/HW4\_Clion/main.cpp, line 363.

**Original Code:**

main.cpp:363

**assert**(igbm(jd\_vg, djher));

main.cpp:328

float djher = iglzqy / ancy; *// 0.1*

**Debug Process:**

1. Debugger shows result is 0 while actually result should be 0,1 according to comment

The result has been rounded, since it has been casted into int,

**Debugged Code** (changes are indicated by red background):

float djher = (float)iglzqy / ancy; *// 0.1*

Memory Erros:

**Memory ERROR 1**

**File:** main.cpp

**Command Line:**

g++ -g main.cpp -o a.out

'/home/cheng\_stark/Desktop/DrMemory-Linux-1.11.17799-1/bin/drmemory' -brief -- a.out --all-operations encrypted\_input.txt secret\_message\_output.txt

**Memory Erros:** 2 memory leaks



**Original Code:**

main.cpp:400

const int hmgo = 25;  
int\*\* ljdsq = new int\*[hmgo];  
int\*\* bdfbc = new int\*[hmgo+1];  
for(int gwcnpy=0; gwcnpy< hmgo; ++gwcnpy) {  
 ljdsq[gwcnpy] = new int[hmgo];  
 bdfbc[gwcnpy] = new int[hmgo+1];  
 for(int wvp\_u=0; wvp\_u< hmgo; ++wvp\_u) {  
 ljdsq[gwcnpy][wvp\_u] = 0;  
 bdfbc[gwcnpy][wvp\_u] = 0;  
 }  
}

**Debug Process:**

1. I noticed that the bdfbc has not been freed, so I added codes to free it
2. After I added free for bdfbc, there are still 90 bytes of leak left. I first tried to directly free thwe memory of the char\* at line275, but that resulted a unaddressible access.
3. I noticed the leak is 90 bytes, that’s the size of the file we read in, so I suspect there are some problems with the read file process.
4. I then scrow down to line 709
5. Added free memory codes

**Debugged Code** (changes are indicated by red background):

added at main.cpp:457

for(int gwcnpy=0; gwcnpy< hmgo; ++gwcnpy) {  
 delete [] bdfbc[gwcnpy];  
}  
delete []bdfbc;

added at main:712

delete[] file\_buffer;

added at main:754

delete[] file\_buffer;

added at main:758

delete[] file\_buffer;

**Memory ERROR 2**

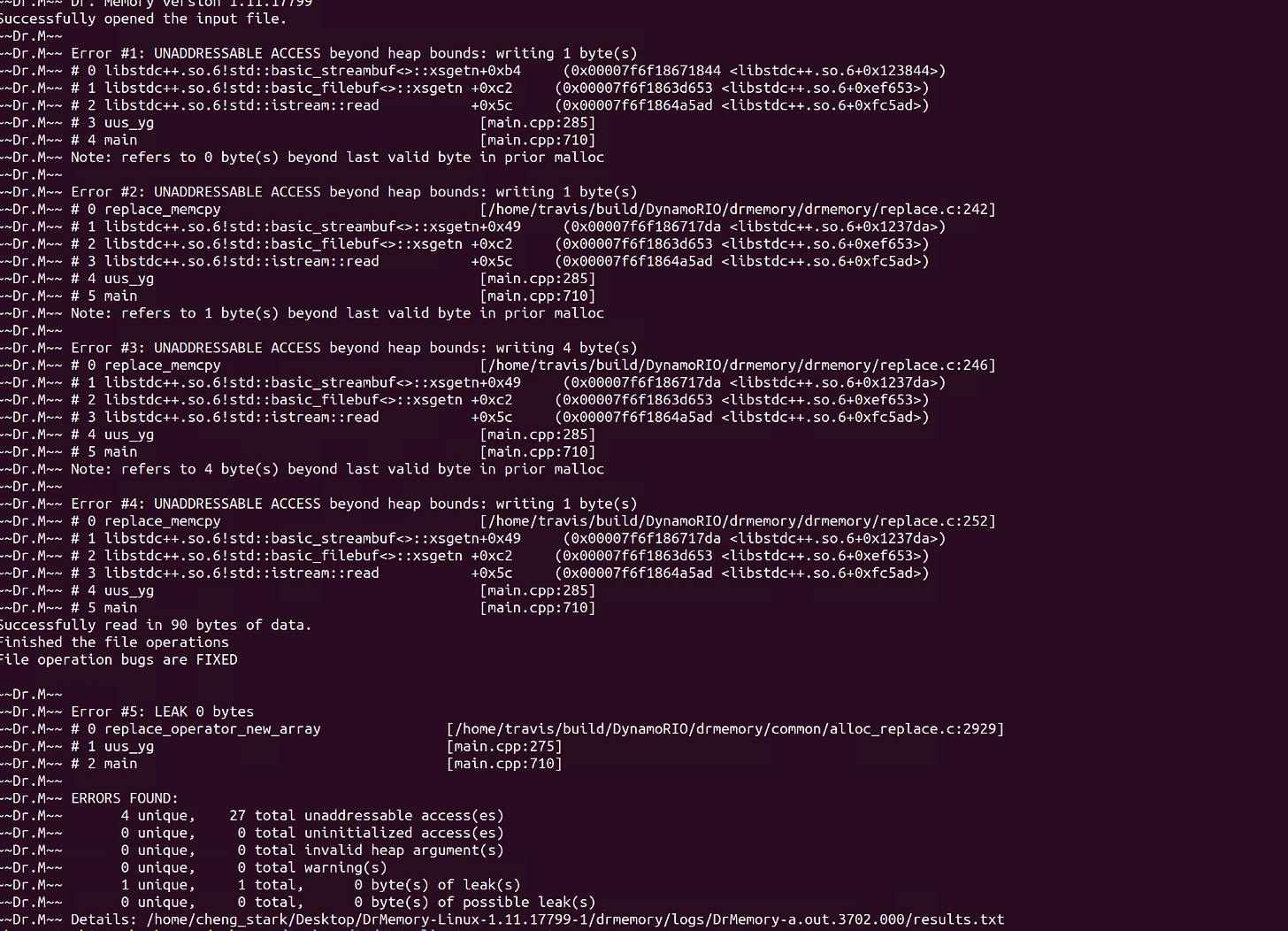
**File:** main.cpp

**Command Line:**

g++ -g main.cpp -o a.out

'/home/cheng\_stark/Desktop/DrMemory-Linux-1.11.17799-1/bin/drmemory' -brief -- a.out --file-operations encrypted\_input.txt secret\_message\_output.txt

**Memory Erros:** 4 unaddressable access(es)



**Original Code:**

main.cpp:272

int wyom;  
  
*// make an array of bytes to hold this information*char\* y\_mxbf = new char[wyom];  
  
*// get the length of the file so we know how much to read  
// this code is from cplusplus.com/reference/istream/istream/read/*ypu\_.seekg(0, ypu\_.end);  
wyom = ypu\_.tellg();  
ypu\_.seekg(0, ypu\_.beg);

**Debug Process:**

1. The int wyom is not initializaed
2. I noticed the wyom is assigned a value later on so I just comment the line wyom and initialize y\_mxbf after the value assignment of wyom

**Debugged Code** (changes are indicated by red background):

*// int wyom;  
  
 // make an array of bytes to hold this information  
  
  
 // get the length of the file so we know how much to read  
 // this code is from cplusplus.com/reference/istream/istream/read/* ypu\_.seekg(0, ypu\_.end);  
 int wyom = ypu\_.tellg();  
 ypu\_.seekg(0, ypu\_.beg);  
  
 char\* y\_mxbf = new char[wyom];