Bug report for hw4

Bug#1

Overall development environment:

MacOS Sierra version 10.12.3 Complier g++

Can you reproduce the problem?

./a.out --arithmetic-operations encrypted\_message.txt output.txt

Describe in words the erroneous behavior:

Assertion failed.

Record the exact text of any error messages.

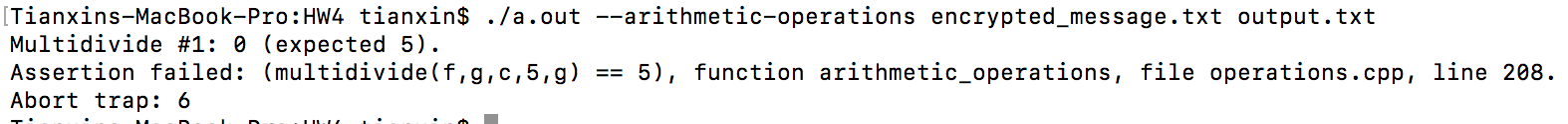
Multidivide #1: 0 (expected 5).

Assertion failed: (multidivide(f,g,c,5,g) == 5), function arithmetic\_operations, file operations.cpp, line 208.

Abort trap: 6

Record the exact output (to the screen or to a ﬁle). How is this diﬀerent from what you expected?

No output here.



Process:

Through the fail information, the assertion of line 208 was failed.

Then, there must are some bugs in the multidivide() function.

Then I found the function multidivide() will return a float number, but take in 5 integers and without converting them to float number. Then the first assertion was failed because of the value type.

add -4 to g

add -4 to e

Bug#2

Overall development environment:

MacOS Sierra version 10.12.3 Complier g++

Can you reproduce the problem?

g++ operations.cpp

Describe in words the erroneous behavior:

Complier warning

Record the exact text of any error messages.

operations.cpp:112:1: warning: control may reach end of non-void function

[-Wreturn-type]

}

Record the exact output (to the screen or to a ﬁle). How is this diﬀerent from what you expected?

No output here.

Bug#3

Overall development environment:

MacOS Sierra version 10.12.3 Complier g++

Can you reproduce the problem?

g++ operations.cpp

Describe in words the erroneous behavior:

Complier warning

Record the exact text of any error messages.

operations.cpp:521:36: warning: expression result unused [-Wunused-value]

for(uint i = 0; i < all.size(); i+1) {

Record the exact output (to the screen or to a ﬁle). How is this diﬀerent from what you expected?

No output here.

Bug#4

Overall development environment:

MacOS Sierra version 10.12.3 Complier g++

Can you reproduce the problem?

g++ operations.cpp

Describe in words the erroneous behavior:

Complier warning

Record the exact text of any error messages.

operations.cpp:532:27: warning: comparison of unsigned expression >= 0 is

always true [-Wtautological-compare]

for(uint i=counter-1; i >= 0; --i) {

Record the exact output (to the screen or to a ﬁle). How is this diﬀerent from what you expected?

No output here.

Bug#5

Input:

Tianxins-MacBook-Pro:HW4 tianxin$ ./a.out --file-operations encrypted\_input.txt secret\_message\_output.txt

Output: Usage: ./a.out operations infile outfile

Couldn't start operations.

Source code:

if(argc == 4) {

std::cerr << "Usage: " << argv[0] << " operations infile outfile" << std::endl;

std::cerr << "Couldn't start operations." << std::endl;

return false;

}

Solution

Change argc==4 to argc!=4

Bug#6

Input:

./a.out --file-operations encrypted\_input.txt secret\_message\_output.txt

Output:

Successfully read in 0 bytes of data.

Assertion failed: (infile.gcount() == length), function file\_operations, file operations.cpp, line 290.

Abort trap: 6

We find that we read in 0 bytes, that is absolutely abnormal

Source code:

int length;

// make an array of bytes to hold this information

char\* buffer = new char[length];

// get the length of the file so we know how much to read

// this code is from cplusplus.com/reference/istream/istream/read/

infile.seekg(0, infile.end);

length = infile.tellg();

infile.seekg(0, infile.beg);

// can't use streaming I/O (the >> operator) because the file is binary data.

// Instead, we'll use the .read() function.

infile.read(buffer, length);

Then we can find that the buffer used length that is not defined.

So I change the position of that line.

Bug#7,8

If(argv==4)

If (ifile)

Bug#9

Tianxins-MacBook-Pro:HW4 tianxin$ ./a.out --list-operations encrypted\_message.txt v.txt

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

-1268210875 letters did not ever appear in the fruit names.

Assertion failed: (\*l1.begin() == 'A'), function list\_operations, file operations.cpp, line 404.

Abort trap: 6

The letter number is abnormal

Source code:

int count;

didn’t initialize.

Bug#10

fruits.erase(++fruit\_itr);

Bug#11

if(\*itr < 'a' || \*itr > 'z') {

continue;

Bug #12

for(char c = 'Z'; c >= 'A'; c--) {

l1.push\_front(c);

}

Bug#13

for(std::list<int>::iterator itr = l500.begin(); itr != l500.end(); ++itr) {

if(\*itr % factor == 0 || \*itr % factor2 == 0) {

l500.erase(itr);

}

Bug#14

Assertion failed: (array[1][2] == -1), function array\_operations, file operations.cpp, line 137.

Abort trap: 6

Bug#15

if((fracpart == 0))

return (int) \*placeholder;

// x is the hypotenuse, need to subtract instead of add

float diffsquares = y\*y - x\*x;

fracpart = modf(sqrt(diffsquares), placeholder);

if((fracpart == 0))

Bug #16

**operations.cpp:529:36: warning: expression result unused [-Wunused-value]**

  for(uint i = 0; i < all.size(); i+1) {

**~^~**

**operations.cpp:540:27: warning: comparison of unsigned expression >= 0 is always**

**true [-Wtautological-compare]**

  for(uint i=counter-1; i >= 0; --i) {

**~ ^  ~**

**operations.cpp:476:7: warning: variable 'counter' is uninitialized when used**

**here [-Wuninitialized]**

      counter++;

**^~~~~~~**

**operations.cpp:472:14: note:** initialize the variable 'counter' to silence this

      warning

  int counter

Bug#17

Thread 2 received signal SIGSEGV, Segmentation fault.

vector\_sum (inVec=...) at operations.cpp:77

77     inVec[i] = inVec[i] + inVec[i-1]

for(uint i=1; i<inVec.size(); ++i) {

inVec[i] = inVec[i] + inVec[i-1];

}

return inVec[inVec.size()-1];

Bug#18

Assertion failed: (v1[2] == 75), function vector\_operations, file operations.cpp, line 458.

Abort trap: 6

int vector\_sum(std::vector<int> &inVec) {

Bug#19

Assertion failed: (v2sum == 55), function vector\_operations, file operations.cpp, line 460.

Abort trap: 6

for(uint i=0; i<10; ++i) {

v2.push\_back(i+1);

}

Bug#20

Assertion failed: (vector\_compare(v1, v4)), function vector\_operations, file operations.cpp, line 507.

Abort trap: 6

bool vector\_compare(const std::vector<int> v1, const std::vector<int> v2) {

bool success = true;

for(uint i=0; i<v1.size(); ++i) {

if(v1[i] > v2[i]) {

success = false;

}

}

return success;

}

->

a

Bug#memory