**--arithmetic-operations**

Bug # 1 int e = b - 3\*a + 5\*c; //32

expected value of e is 32 but got 36

fix- int e = b - 3\*a + 4\*c;

also fixes the value of g

Bug #2 int h = (f/c) / a;

Expected value of 3 but got 2

Fix- int h = ceil(double((f/c)) / a);

Also fixes the values of m and n

Bug #3 int p = (f / e) - h; // -1

Expected value -1 but got 0

Fix- int p = ((f / e)-1) - h; // -1

Bug #4- int r = g + m + p + n;

Expected value -8, got -7

Fix- int r = g + m + p + n + -1;

Bug#5- float s = a / f;

Expected value .1 got 0

float s = float(a) / f;

Bug #6 float f = ((((numerator / d1) / d2) / d3) / d4);

Need to cast to float in order to get correct numbers when dividing, instead of int/int, need float/int

float f = ((((float(numerator) / d1) / d2) / d3) / d4);

**--file-operations**

Bug #1- if(argc == 4)

Outputted - Usage: ./test.exe operations infile outfile

Couldn't start operations.

Fix - if(argc < 3 )

This code is also unnessary because this was checked in main, but this is the least character solution

Bug #2 - Memory

~~Dr.M~~ Error #1: LEAK 131 bytes

~~Dr.M~~ # 0 replace\_operator\_new\_array [/drmemory\_package/common/alloc\_replace.c:2928]

~~Dr.M~~ # 1 file\_operations [/Users/Mike/Dropbox/Data Structures/hws/hw4code/hw4code/operations.cpp:75]

~~Dr.M~~ # 2 main [/Users/Mike/Dropbox/Data Structures/hws/hw4code/hw4code/operations.cpp:608]

~~Dr.M~~

~~Dr.M~~ ERRORS FOUND:

~~Dr.M~~ 0 unique, 0 total unaddressable access(es)

~~Dr.M~~ 0 unique, 0 total uninitialized access(es)

~~Dr.M~~ 0 unique, 0 total invalid heap argument(s)

~~Dr.M~~ 0 unique, 0 total warning(s)

~~Dr.M~~ 1 unique, 1 total, 131 byte(s) of leak(s)

~~Dr.M~~ 0 unique, 0 total, 0 byte(s) of possible leak(s)

~~Dr.M~~ Details: /Users/Mike/Dropbox/Data Structures/DrMemory/DrMemory-MacOS-1.11.0-2/drmemory/logs/DrMemory-test.exe.36909.000/results.txt

singlet-81:hw4code Mike$

Fix – delete [] file\_buffer //at end of main

**--array-operations**

Bug #1- for(int x=1; x<=size; ++x) {

array[x] = new int[size];

for(int y=1; y<=size; ++y) {

array[x][y] = 0;

}

}

fix- for(uint x=0; x<size; ++x) {

array[x] = new int[size];

for(uint y=0; y<size; ++y) {

array[x][y] = 0;

}

}

Bug #2

for(uint x=1; x>=size; ++x) {

for(uint y=1; y>=size; ++y) {

array[x][y] = pythagoras(x, y);

}

}

Fix –

for(uint x=0; x<size; ++x) {

for(uint y=0; y<size; ++y) {

array[x][y] = pythagoras(x, y);

}

}

Bug#3 same as 1 and 2 just different spot

Bug #4 double\* placeholder; //in pythagorus

Fix double place holder

Because modf takes a double and a &double

Bug #5 return (int) \*placeholder;

return (int) sqrt(sumsquares); //in pythagorus

Bug #6 segmentation fault 10

int\*\* tmp\_ptr = array;

for(int x = 1; x <= size; ++x, ++tmp\_ptr) {

int\* tmp\_ptr2 = \*tmp\_ptr;

for(int y = 1; y <= size; ++y, ++tmp\_ptr) {

int tmp = \*tmp\_ptr2;

// pad single-digit numbers with a space so it looks nice

// ain't nobody got time for <iomanip>

std::string maybe\_space = ((tmp < 10 && tmp >= 0) ? " " : "");

std::cout << maybe\_space << \*tmp\_ptr2 << " ";

Fix- for(int x = 0; x < size; ++x) {//Bug #6

int\* tmp\_ptr2 = \*(tmp\_ptr+x);

for(int y = 0; y < size; ++y) {

int tmp = \*(tmp\_ptr2+y);

// pad single-digit numbers with a space so it looks nice

// ain't nobody got time for <iomanip>

std::string maybe\_space = ((tmp < 10 && tmp >= 0) ? " " : "");

std::cout << maybe\_space << tmp << " ";

**--vector-operations**

Bug #1

Assertion failed: (v1sum == 175), function vector\_operations, file operations.cpp, line 420.

Abort trap: 6