Environment Overview

Key	Value
Language Standard	C++ 11
Compiler	g++ (Debian 6.3.0-18+deb9u1) 6.3.0 20170516
gdb	GNU gdb (Debian 7.12-6) 7.12.0.20161007-git
Valgrind	3.14.0
IDE	CLion 2018.3.3 Build #CL-183.5153.40, built on January 9, 2019
OS	Windows 10 Education 1803 17134.523

Bugs

`--arithmetic-operations`

First, I tried running directly and catch the error message.

Issue 1

```
int uzlaki(): Assertion `qaut(wbijf,gqrre,ymvu,5,gqrre) == 5' failed.
```

Possible reason

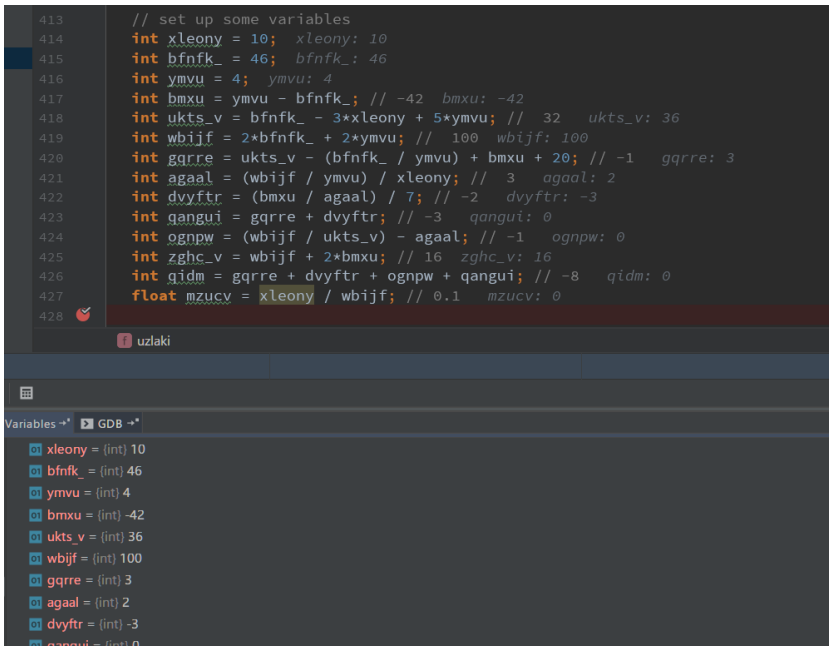
First, I chose to look into method `gaut`:

```
float wlid = ((((hhyo / qykrae) / vuwpz) / rtazkk) / tval);
```

The result of integer division used in a floating point context may cause possible loss of precision.

- Casting float `(float)` add the front of each variable.

However, the number given cannot reproduce such result, so I set a breakpoint in `gdb` and following result has shown:



It looked like the result did not match with the annotation.

```
int ukts_v = bfnfk_ - 3*xleony + 5*ymvu; //  32
```

- It should be `4 * ymvu`. Previously it is `36`

```
int agaal = (wbijf / ymvu) / xleony; //  3
```

- I added `+ ((wbijf / ymvu) % xleony > 0)` at the back since if we want `100 / 4 / 10` (result 2.5) == 3, we need to do `ceil`

Some similar issues are ignored.

`--file-operations`

Issue when opening file

```
Usage: {PATH} operations infile outfile
Couldn't start operations.
```

Possible reasons

Method `teqd` handled file open. I searched error string for code location.

```
if(argc == 4) {
  ...
```

```
  return false;
}
```

- `argc` should be 4, change `if` condition to `!=`

Issue when doing operation

```
bool teqd(int, char**, char*&, int&): Assertion `qqioi.gcount() != nciy' fai
led.
```

Possible reason

Assertion tried to check data length.

```
assert(qqioi.gcount() != nciy);
```

- the assertion condition should be `==` since `nciy` is the length of the data read in.

`--array-operations`

Issue 1

```
Process finished with exit code -1
```

No explicit error message appeared.

Procedures

I tried finding where `array-operations` called in `main`:

```
if(ops == "--array-operations" || ops == "--all-operations") {
  records[1] = xujz();  ...
}
```

I traced function `xujz()`, and set breakpoint at `line 653`, and SIGSEGV (Segmentation fault) caught at `line 650`.

After inspecting the first for-loop, I consider that:

`xsdus[yphmro+1]` will point to `NULL` since it is not init at the first for-loop.

- Since it did not make sense to `+1` in this for-loop (will be out of bound), I removed them;
- And if created an array with `[num]`, the range should be `0-24`; the later function uses the index directly `new` and this function is about to store a pair of coordinate, guess that `neug` should have the same index as `xsdus`: I have changed index accordingly.

However, the program still returned `-1`. I set a breakpoint at `line 666` found Segmentation fault at `line 657`.

```
assert(xsdus[-1][-1] == 0);
```

- Index `-1` did not make sense. Since it is testing the corner cases, I change them into `ohmgu - 1`

Issue when store number

I re-run the program after fixation

```
int xujz(): Assertion `xsdus[1][2] == -1' failed.
```

Procedures

I set up a breakpoint right after the loop and examine the expression `xsdus[1][2]`, which return `0`

- The breakpoint was skipped, it is found that the for-loop condition is wrong: I changed from `>=` to `<=`

I stepped into the source of the result provided (function `ofxtfp`)

There are several Issues in this method:

- The if condition should not be assignment, change accrodingly
- Only two conditions considered (missing `ribzwv` is hypotenuse)
- Return `-1` when no found was not considered.

`--vector-operations`

Issue 1

```
Process finished with exit code -1
```

No explicit error message

Procedures

I found method call in main:

```
records[2] = taysxt();
```

I traced into this method and randomly set a breakpoint within the function (`line 147`); and nothing was breaking.

I stepped over; a Segmentation Fault found at `line 393` in method `wwqub` . From the note, it says it would modify the vector pass in.

- I changed parameters from copy into reference so the modification can be used later.

The index cannot be equal to the size of the vector

- I use `<` instead of `<=` in for-loop.

Since for-loop will access element at `i` and `i - 1`

- the initial condition should be 1 instead of 0.
- The return condition should be `size() -1`

Issue 2

```
Now counting numbers divisible by 3
```

The program seemed stuck here (running after a while without any output).

Procedures

I searched the string printed and found that the message printed at `line 222` . A for-loop is right after it.

```
for(uint fqfkg = 0; fqfkg < gqszp.size(); fqfkg+1) {
```

- The plus-one (`fqfkg + 1`) result not used: I changed it into `fqfkg+=1`

`--list-operations`

Issue 1

Found Segmentation Fault at `line 283`

Procedures

I doubted that Segmentation Fault should be a kind of out-of-bound issue, so I check the for-loop:

```
nvlahj.erase(swem);
```

- `erase` will return the next element and should be used to replace current iterator.

Issue 2

```
elderberry ...
-1224707885 letters did not ever appear in the fruit names.
int zbna(): Assertion `*fkciz.begin() == 'A'' failed.
```

Procedures

- The int printed looks strange, it is not initialized.

I compared my output with `expected_output.txt`, the fruit list is not complete.

```
// remove non-fruits from the list
...
bijqwy.erase(++bkrm);
```

- `erase` should remove target rather than `++target`

I used GDB and set a breakpoint at `line 361`, print out what is inside `fkciz`. It is noticed that the list start with `Z`. The order of the list is wrong when initialize.

- I reversed the for-loop for upper-case letter first and then lower-case letter.
- I replaced `push_front` with `push_back`.