

Task1

1.1

```
Program received signal SIGFPE, Arithmetic exception.
0x000055555555204 in improve (a=-2, b=0) at marks.cpp:17
17         return ((double)(a / b)) * 100;
(gdb) list
12         return 0;
13     }
14
15     int improve(int a, int b)
16     {
17         return ((double)(a / b)) * 100;
18     }
(gdb) where
#0  0x000055555555204 in improve (a=-2, b=0) at marks.cpp:17
#1  0x0000555555551dd in main () at marks.cpp:11
(gdb) frame 1
#1  0x0000555555551dd in main () at marks.cpp:11
11         cout << improve(mark, highest);
(gdb) list
6
7     int main() {
8         int mark = 59, highest = 87;
9         cout << improve(mark, highest);
10        mark = -2; highest = 0;
11        cout << improve(mark, highest);
12        return 0;
13    }
14
15    int improve(int a, int b)
(gdb) print highest
$1 = 0
(gdb) |
```

- 3.) Floating point error (SIGFPE), on line 17 with the values of a = -2, b = 0.
- 4.) The return line of the improve function.
- 5.) The main function is where the improve function was called and the parameters were passed in from.
- 9.) The crash occurred due to trying to divide by 0.

## 1.2

```
==12642== Memcheck, a memory error detector
==12642== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==12642== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==12642== Command: ./Main
==12642==
==12642== Invalid write of size 4
==12642==    at 0x10916B: capture() (capture.cpp:4)
==12642==    by 0x109180: main (capture.cpp:9)
==12642== Address 0x4da8ca8 is 0 bytes after a block of size 40 alloc'd
==12642==    at 0x483C583: operator new[](unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==12642==    by 0x10915E: capture() (capture.cpp:3)
==12642==    by 0x109180: main (capture.cpp:9)
==12642==
==12642== HEAP SUMMARY:
==12642==    in use at exit: 40 bytes in 1 blocks
==12642==    total heap usage: 2 allocs, 1 frees, 72,744 bytes allocated
==12642==
==12642== 40 bytes in 1 blocks are definitely lost in loss record 1 of 1
==12642==    at 0x483C583: operator new[](unsigned long) (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==12642==    by 0x10915E: capture() (capture.cpp:3)
==12642==    by 0x109180: main (capture.cpp:9)
==12642==
==12642== LEAK SUMMARY:
==12642==    definitely lost: 40 bytes in 1 blocks
==12642==    indirectly lost: 0 bytes in 0 blocks
==12642==    possibly lost: 0 bytes in 0 blocks
==12642==    still reachable: 0 bytes in 0 blocks
==12642==    suppressed: 0 bytes in 0 blocks
==12642==
==12642== For lists of detected and suppressed errors, rerun with: -s
==12642== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
```

3.) The process ID = 12642.

4.) Invalid write. The program wrote to some memory that it should not have due to writing past the allocated block (overflow).

5.) It tells where the error occurred as well as the calls leading up to it. i.e. The error is at line 4, in the capture function, and was called by the main function, at line 9.

6.) 40 bytes has been lost as 1 int is 4 bytes and the program allocated memory for 10 int, hence 40 bytes, and during the runtime of the program, that allocated block of memory was not freed.

7.) To fix this issue, you must simply free the allocated memory before the program exits. i.e. delete marks;