

Errors in Code...

- **Problem:**
We want to avoid certain values for a variable.
- **Question:**
How?

General Hints

- `assert(expr);`
 - `expr` is `true`: nothing happens
 - `expr` is `false`: stop program

Example

Problem: Some inputs are dangerous.

```
#include <iostream>

int main () {
    int a;
    std::cin >> a;
    int b;
    std::cin >> b;

    // Output: a/b
    std::cout << a/b << "\n";

    return 0;
}
```

Problem for:

b == 0

Example

Problem: Some inputs are dangerous.

assert ensures

b != 0

```
    return 0;  
}
```

Example

Problem: Some inputs are dangerous.

```
#include <iostream>
#include <cassert>

int main () {
    int a;
    std::cin >> a;
    int b;
    std::cin >> b;
    assert(b != 0);

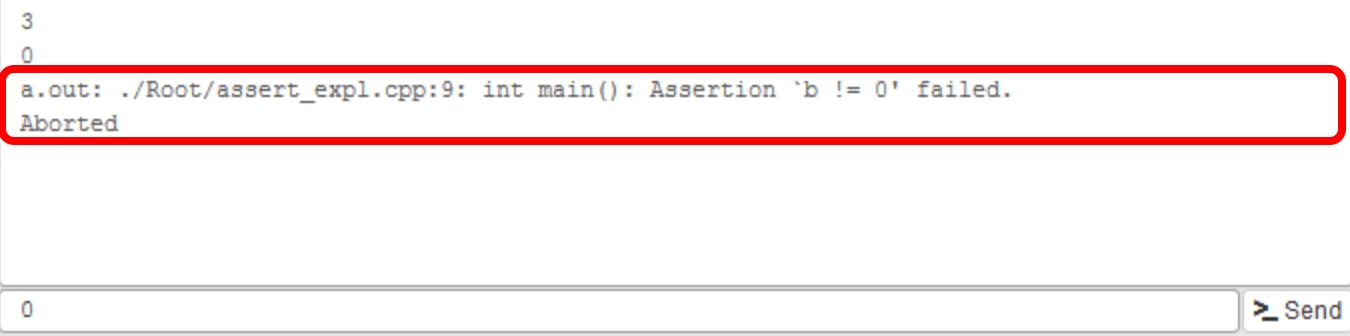
    // Output: a/b
    std::cout << a/b << "\n";

    return 0;
}
```

Example

Problem: Some inputs are dangerous.

```
#include <iostream>
```



A screenshot of a terminal window. The input was "3\n0". The output was "a.out: ./Root/assert_expl.cpp:9: int main(): Assertion `b != 0' failed.\nAborted". The entire output line is highlighted with a red rectangle. At the bottom, there is an empty input field containing "0" and a "Send" button.

```
3
0
a.out: ./Root/assert_expl.cpp:9: int main(): Assertion `b != 0' failed.
Aborted
```

```
// Output: a/b
std::cout << a/b << "\n";

return 0;
}
```

assert - Why?

- Still an easy example...
- So **why** and **where** is assert useful?

assert - Why?

- Still an easy example...
- So **why** and **where** is assert useful?
 - Long programs: for overview
 - User-Inputs required: for safety
 - Multiple programmers: for safety