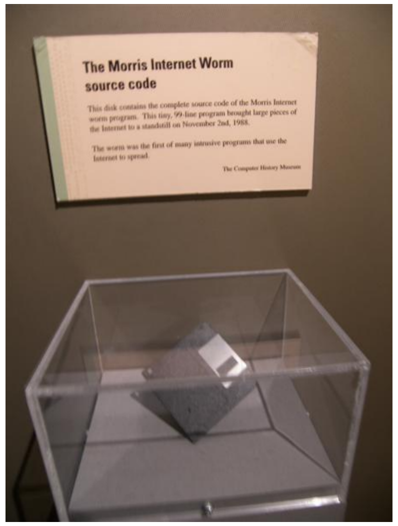
institute of CODING

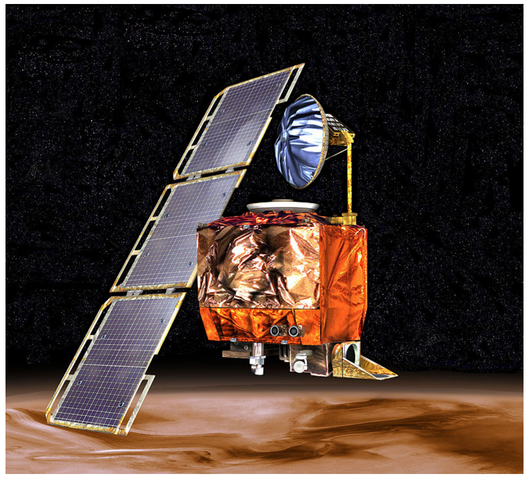
COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE

Queen Mary
University of London

The Top 10 bugs

www.cs4fn.org/bugs/





BUG 10: Divide by Zero

The USS Yorktown heralded a new generation of “smart” ship. Its propulsion system failed leaving it "dead in the water” for 3 hours after it tried to divide by zero crashing every computer on the ship’s network.

BUG 9: Arithmetic Overflow

Keep adding to an integer variable and you run out of bits. Suddenly you have a small number not the bigger one expected.

This was one bug in the Therac-25 radiation therapy machine that killed patients.

The Boeing 787 Dreamliner had the same problem. Fly for more than 248 days and it would switch off.

BUG 8: Timing problems

AT&T lost $60 million the day the phones died (all of them). It was a result of changing a few lines of working code. Things happened too fast for the program. The telephone switches reset but were told they needed to reset again before they'd finished, … and so on.

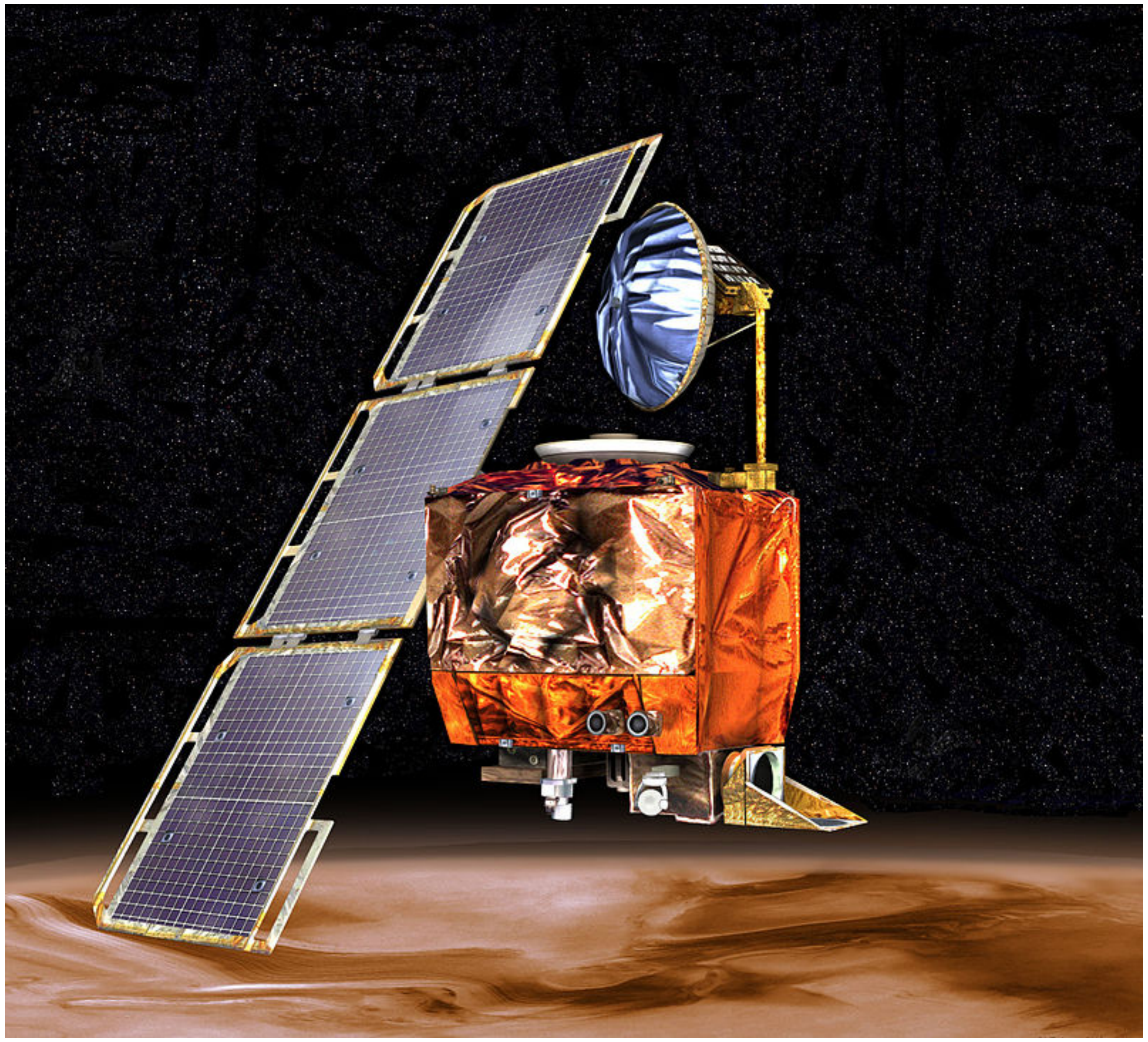
BUG 6.99999989:
Wrong numbers in a lookup table

Intel's Pentium chip turned out not to be able to divide properly. It was due to a wrong entry in a lookup table. Intel set aside $475 million to cover replacing the flawed processors. Some chips were turned in to key rings.

Konstantin Lanzet CC BY-SA 3.0

BUG 6: Wrong units

The Mars Climate Orbiter spent 10 months getting to Mars …where it promptly disintegrated. It passed too close to the planet’s atmosphere. The programmers assumed numbers were in pound-force seconds when they were actually in newton-seconds.



BUG 5: Non-terminating loop

The spinning pizza of death is common. Your computer claims to be working hard, and puts up a progress symbol like a spinning wheel…forever. There are lots of ways that this happens. The simple version is that the program has entered a loop in a way that means the test to continue is never false.

BUG 4: Storing a big number in a small space

The first Ariane 5 rocket exploded at a cost of $500 million 40s after lift-off. Despite $7 billion spent on the rocket, the program stored a 64 bit floating point number in to a variable that could only hold a 16 bit integer.

DLR German Aerospace Center CC BY 2.0

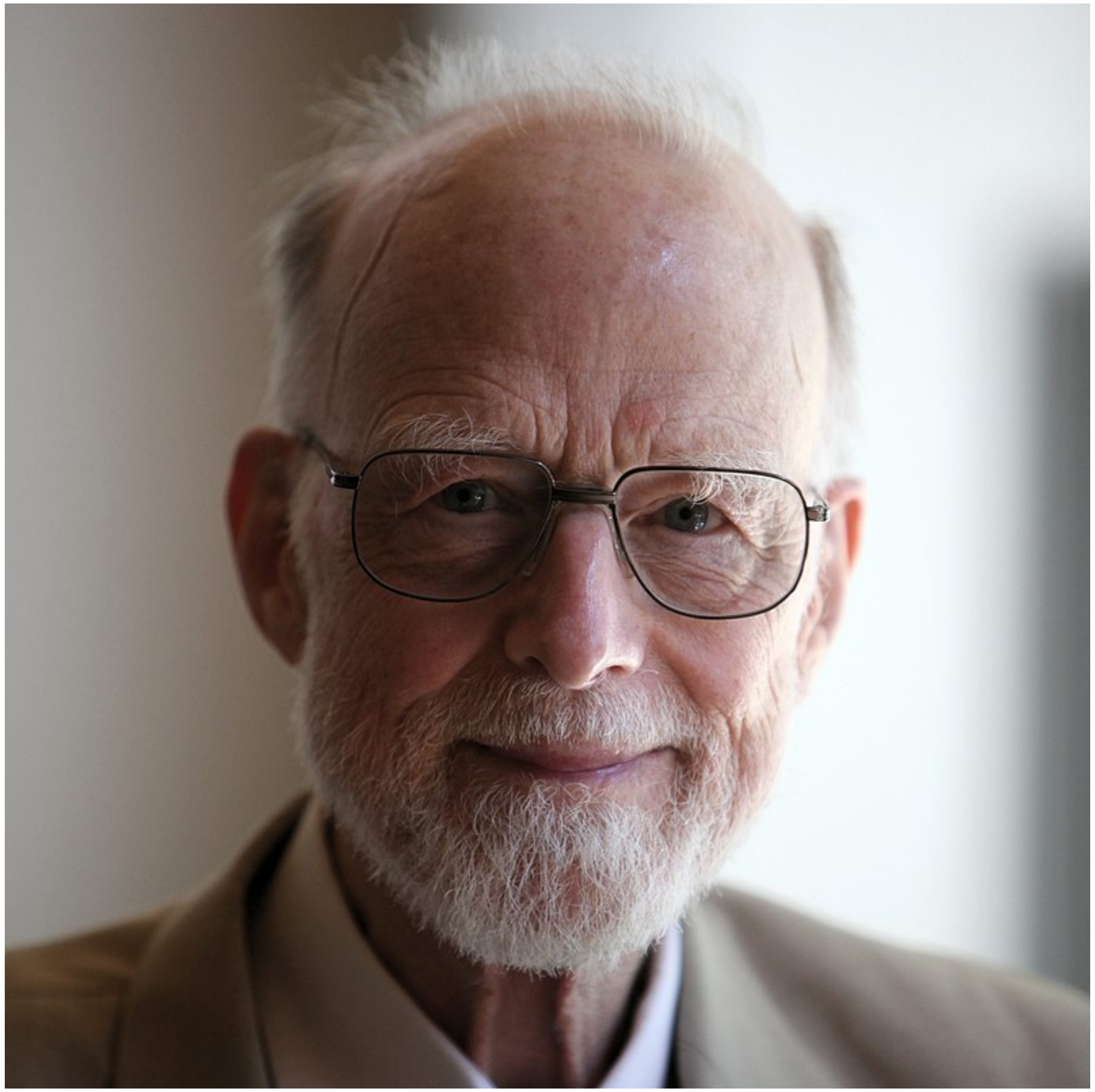
BUG 3: Memory leak

Memory leaks (forgetting to free up space when done with) are responsible for many computer problems. The Firefox browser had one. It was infamous because Firefox (implausibly) claimed their program had no memory leaks.

The Mozilla Foundation CC BY 3.0

BUG 2: Following a null pointer

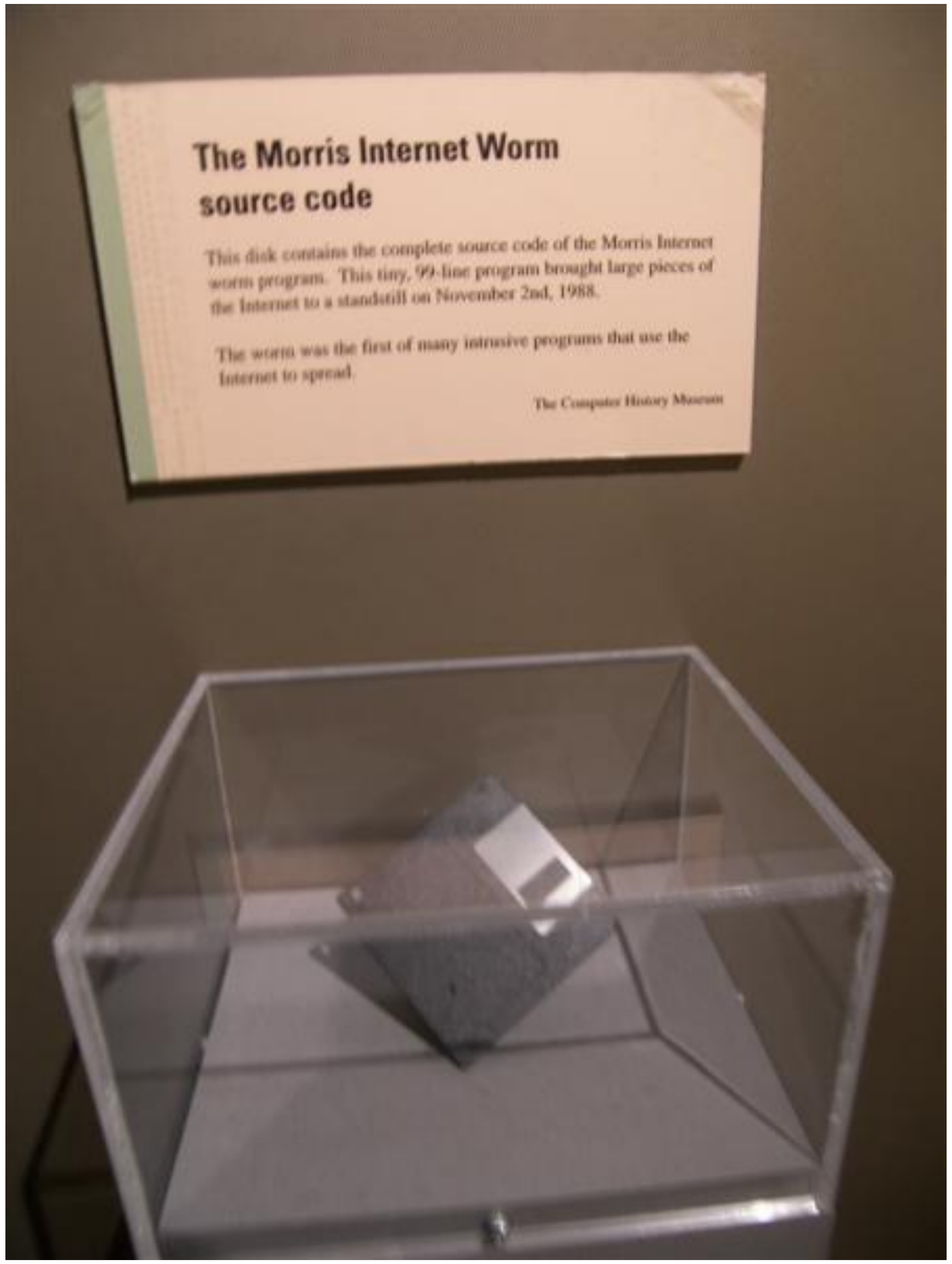
Tony Hoare who invented the null pointer (a pointer that points nowhere) called it his “billion-dollar” mistake because programmers struggle to cope with it. **Null pointer bugs** crash computers, give hackers ways in and generally cause chaos.



Photograph by Rama, Wikimedia Commons, Cc-by-sa-2.0-fr

BUG 1: Buffer overflow

The Morris Worm, an early Internet worm, came close to shutting down the Internet. It used a **buffer overflow** bug in network software to move from computer to computer shutting them down.


The Morris Internet Worm
source code
This disk contains the complete source code of the Morris Internet worm program. This tiny, 99-line program brought large pieces of the Internet to a standstill on November 2nd, 1988.
The worm was the first of many intrusive programs that use the Internet to spread.
The Computer History Museum

Go Card USA from Boston, USA: CC BY-SA 2.0

BUG 0: Running off the end of an array

Arrays in many languages start from position 0. This means the last position is one less than the length of the array. Get it wrong… as every novice programmer does at some point … and you run off the end.

Oddly (in Europe), we have no problem in a lift pressing 1 to go to the second floor up.

In other situations, count from 0 and you do one too many things.

institute of CODING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE

Queen Mary
University of London

Download these slides from

teachinglondoncomputing.org/bugs/