`program.py` code review

Originally developed by Mike Jackson

EPCC, The University of Edinburgh

michaelj@epcc.ed.ac.uk

With acknowledgement to Greg Wilson (Software Carpentry), James Hetherington (University College London), Andrew Turner (University of Edinburgh) and EPCC MSc students of 2016.

5 October 2017

1 A Python powers of two program

The following section shows the source code for a small Python program, `program.py`. It takes a single non-negative integer as a command line argument and prints out powers of two that, when added together, give the original integer. For example, running the program with 27 gives:

```
$ python program.py 27
16
8
2
1
```

These are the powers of 2 of 27, as 16 + 8 + 2 + 1 = 27.

Other examples include:

```
$ python program.py 1
1

$ python program.py 2
2
```

```
$ python program.py 3
2
1

$ python program.py 255
128
64
32
16
8
2
1

$ python program.py 256
256

$ python program.py 257
256
1
```

If given a value less than or equal to zero, then there is no output:

```
$ python program.py 0
```

2 program.py

```
#!/usr/bin/python
import sys

def p2(sum):
    ps = []
    x = 1
    while (sum > 0):
        if (sum % 2):
            ps.insert(0, x)
        x = x * 2  # Multiply by 2. Is this a bug?
        sum = sum >> 1 # Do a shift.
    # Print the powers.
    for x in ps:
        print x
    return ps

if __name__ == '__main__':
    # TODO Convert sys.arvg[1] into an integer.
    value = int(sys.argv[1])
    p2(value)
```

3 Code review

Look at this program and review its code. Look for any issues, or problems you see, or improvements you could make, to do with:

- How easy it is to read.
- How easy it is to understand.
- How the code is structured or designed.
- How the code is documented.
- Anything else you think is a problem.

Note down all the issues you find, explanations as to why these are issues, and how these issues could be fixed i.e. everything you think is wrong about the code and how it could be fixed.

This is not a code development exercise, there is no need to write any code. Nor is this a debugging exercise, the code has no bugs (that the author is aware of).

Though the code is written in Python, some of the problems it has are problems that are universal, and would equally be problems if the code were written in C, C++, Fortran or Java.