Input/output in Fortran

Victor Eijkhout and Carrie Arnold and Charlie Dey

Fall 2017

I/O commands

- `print` simple output to terminal
- `write` output to terminal of file ('unit')
- `read` input from terminal or file
- `open`, `close` for files and streams

Array printing

- `print *,A` prints whole array, column-major
- Implicit do loops:

  ```
  print *,( A(i,i),i=1,n)
  ```

  Can also be nested.

Formats

- Fine control of input/output.
- Direct use in print statement:

  `print '(a6,3f5.3)',"Result",x,y,z`
- Format statement:

```
  print 10,"result:",x,y,z
10 format('(a6,3f5.3)')
```

Format specifiers

- ‘`a`*n*’ specifies a string of *n* characters. If the actual string is longer, it is truncated in the output.
- ‘`i`*n*’ specifies an integer of up to *n* digits. If the actual number takes more digits, it is rendered with asterisks.
- ‘`f`*m.n* specifies a fixed point representation of a floating point number, with *m* total positions (including the decimal point) and *n* digits in the fractional part.
- `e`*m.n* Exponent representation.

Putting a number in front of a single specifier indicates that it is to be repeated.

Repeats and line breaks

- If `abc` is a format string, then `10(abc)` gives 10 repetitions. There is no line break.
- If there is more data than specified in the format, the format is reused in a new print statement. This causes line breaks.
- The / (slash) specifier causes a line break.

Exercise 1

Use formatted I/O to print the number $0 \cdots 99$ as follows:

```
 0  1  2  3  4  5  6  7  8  9
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59
60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79
80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99
```