Looping in Fortran

Victor Eijkhout, Susan Lindsey

Fall 2019

Indexed Do loops

```
integer :: i

do i=1,10
  ! code with i
end do
```

You can include a step size (which can be negative) as a third parameter:

```
do i=1,10,3
  ! code with i
end do
```

While loop

The while loop has a pre-test:

```
do while (i<1000)
  print *,i
  i = i*2
end do
```

Exit and cycle

Loop without counter or while test:

```
do
  call random_number(x)
  if (x>.9) exit
  print *,"Nine out of ten exes agree"
end do
```

Skip rest of current iteration:

```
do i=1,100
  if (isprime(i)) cycle
  ! do something with non-prime
end do
```

Exercise 1

Read an integer and determine whether it is prime by testing for the smaller numbers whether they are a divisor of that number.

Print a final message

```
Your number is prime
```

or

```
Your number is not prime: it is divisible by ....
```

where you report just one found factor.

Implied do loops

```
print *,(2*i,i=1,20)
```

You can iterate multiple expressions:

```
print *,(2*i,2*i+1,i=1,20)
```

These loops can be nested:

```
print *,( (i*j,i=1,20), j=1,20 )
```

Exercise 2

Use the implied do-loop mechanism to print a triangle:

```
1
2 2
3 3 3
4 4 4 4
```

up to a number that is input.