

## Fortran 77 : Arrays

The value of computers to modern society comes from their ability to store and efficiently process large volumes of information. Arrays<sup>1</sup> are simple data structures that can be used to store large amounts of data. Like all other programming languages, Fortran has its own means of defining and manipulating arrays.

### Declaration of Arrays

One method of declaring arrays is by using the dimension statement. For example the following commands

```
DIMENSION A(2)
```

```
DIMENSION N(3,2)
```

set up an array A with two components and an array N with six components. Since there is no declaration of type then Fortran would normally revert to its intrinsic definitions; A is an array of real numbers and N is an array of integers. Since there is no declaration of precision then Fortran will normally use standard precision of 4 byte real numbers and integers.

It is normally good practice in computer programming to be as clear as you can be about how the computer program works. Fortran 77 has the facility of declaring the type and precision when declaring arrays and it is suggested that this should be used whenever any possible misunderstanding may arise.

For example the above declarations could be made in the following way, which explicitly states the type.

```
REAL A(2)
```

```
INTEGER N(3,2)
```

If we also want to specify the precision then, we could use the declarations

```
REAL*4 A(2)
```

```
INTEGER*4 N(3,2)
```

Arrays of any data type may be declared and a number of precisions. For example

```
COMPLEX*16 Z(10)
```

declares an array of ten double precision complex numbers and

```
LOGICAL L(5)
```

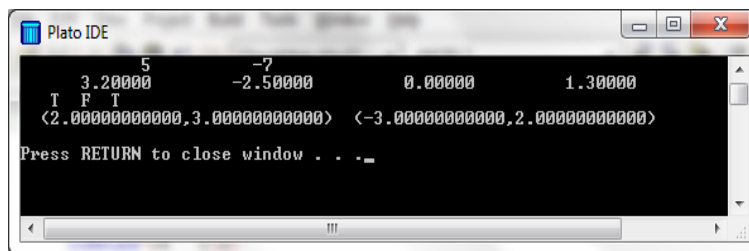
declares an array of five logical variables.

---

<sup>1</sup> [Arrays](#)

The following Fortran program and output demonstrate the setting up and print out of arrays. Note the used of the DATA statement to input data into an array.

```
C Arrays
PROGRAM ARRAYS
INTEGER  N(2)
REAL*4  A(2,2)
LOGICAL  L(3)
COMPLEX*16 C(2)
INTEGER  I,J
DATA (N(I),I=1,2) / 5, -7 /
DATA ((A(I,J), I=1,2), J=1,2) / 3.2, -2.5, 0.0, 1.3 /
DATA (L(I),I=1,3) / .TRUE., .FALSE., .TRUE. /
DATA (C(I),I=1,2) / DCMPLX(2,3), DCMPLX(-3,2) /
WRITE (*,*) N
WRITE (*,*) A
WRITE (*,*) L
WRITE (*,*) C
END
```



### Accessing Arrays

Arrays are effectively subscripted variables. Each component of an array can be accessed for assignment of values to the array or to find the value of components of the array.

```
C Accessing arrays
PROGRAM ACCARR
INTEGER N(2)
REAL*4 A(2,2)
N(2)=5
A(1,2)=7.0
A(2,2)=-3.0
WRITE (*,*) N(2)
WRITE (*,*) A(1,2),A(2,2)
END
```

