

## Fortran 77: Loops

If a piece of code needs to be executed repeatedly then a loop structure can be used. In most programming languages, using loops is the means to getting a computer to do any significant *work*.

There is only one defined loop structures in Fortran 77; the DO loop. The DO loop can only be used in cases when the number of loops can be defined on entry to the loop. Alternative loop structures can be created by using a combination of the IF and GOTO statements.

### The DO Loop

The syntax of the DO loop is as follows:

```
DO <label>, <loop control variable>=<initial value>,<final value>,<step size>
```

```
<statement1>
```

```
<statement2>
```

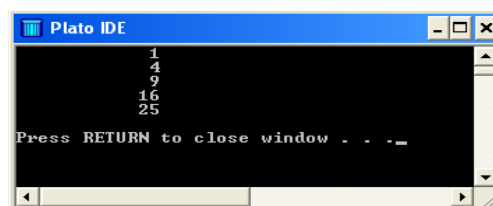
```
.....
```

```
<statementN>
```

```
<label> CONTINUE
```

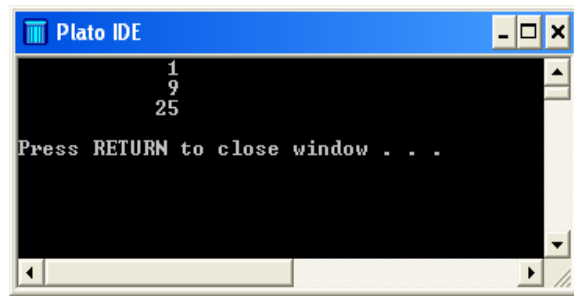
In this example J takes the values 1,2,3,4,5 and J\*J is output for each values of J.

```
C *****
C DO LOOP
C *****
PROGRAM DOLOOP
INTEGER J
DO 10 J=1,5
WRITE(*,*) J*J
10 CONTINUE
STOP
END
```



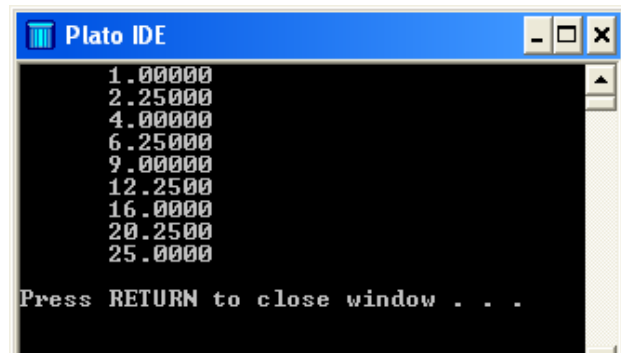
In the above example the number step in J from one execution of the loop to the next is one; this is the default value. We can have steps of alternative size. For example in the next piece of code the step size is 2 (placed after the 1 and the 5). In this code J takes the values 1, 3, 5.

```
C *****
C DO LOOP 2
C *****
PROGRAM DO2LOOP
INTEGER J
DO 10 J=1,5,2
WRITE(*,*) J*J
10 CONTINUE
STOP
END
```



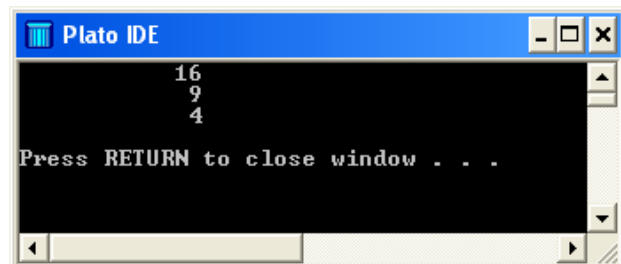
The step size in a loop does not have to be a whole number. In the following example the step size is 0.5.

```
C *****
C DO LOOP 3
C *****
PROGRAM DO3LOOP
REAL X
DO 10 X=1.0,5,0.5
WRITE(*,*) X*X
10 CONTINUE
STOP
END
```



The step size in a loop does not have to be positive. In the following example the step size is -1.

```
C *****
C DO LOOP 4
C *****
PROGRAM DO4LOOP
INTEGER J
DO 10 J=4,2,-1
WRITE(*,*) J*J
10 CONTINUE
STOP
END
```

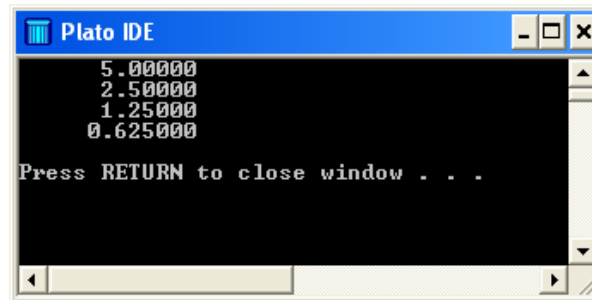


The <initial value>, <final value>, and <step size> may all be variables, but they cannot be changed inside the loop.

### Mock *while* loop

A while loop continues while a condition is met. Fortran 77 does not have an explicit while loop, but one can be created using the GOTO statement. For example, in the following loop, X is continued to halve while X is greater than 1 and the loop terminates.

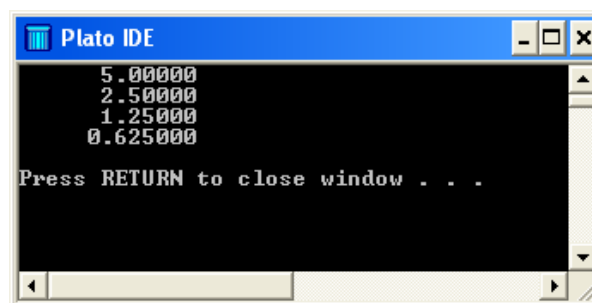
```
C *****
C WHILE LOOP
C *****
PROGRAM WHLOOP
REAL X
X=10.0
10 IF (X.GT.1) THEN
  X=X/2
  WRITE(*,*) X
  GOTO 10
ENDIF
STOP
END
```



### Mock *repeat* loop

A repeat loop continues until a condition is met. Fortran 77 does not have an explicit repeat loop, but one can be created using the GOTO statement. For example, in the following loop, X is continued to halve until X is no longer greater than 1 and the loop terminates.

```
C *****
C REPEAT LOOP
C *****
PROGRAM RELOOP
REAL X
X=10.0
10 CONTINUE
  X=X/2
  WRITE(*,*) X
  IF (X.GT.1) GOTO 10
STOP
END
```



### Good Practice Tip

For readability, always indent the statements inside a loop.