

arithmetic is considerably faster than floating point arithmetic, you can often speed up a program by replacing floating point FOR commands with integer FOR commands. For example, the command

```
FOR I = 5 TO 6.5 STEP .03
```

could be replaced by

```
FOR I% = 500 TO 650 STEP 3
```

Like GOSUB, FOR uses memory space to store information about the FOR loop, and the space is reclaimed when you exit from the loop. An OM (Out of Memory) error will occur if insufficient space remains.

The following example shows how useful FOR loops can be in printing tables.

Example:

Charles Squaro works for a company which makes cubic boxes with sides between 5 and 30 inches. He wants to know how much wood is used and the resulting volume for each type of box, and wants a table he can refer to when ordering. Since the volume of a cube is the cube of its side, and the surface area of a cube is six times the square of its side, the following program prints the desired information.

```
NEW
OK
10 PRINT "SIZE", "AREA", "VOLUME"
20 FOR I = 5 TO 30
30 PRINT I, I*I*6, I*I*I
40 NEXT I
```

ON / GOTO and ON / GOSUB

The ON / GOTO and ON / GOSUB commands let you use the value of a variable or formula to choose which of a group of tasks to perform. For example, execution of the command

```
ON VAR GOTO 10,20,30,40
```

transfers control to line 10 if the value of VAR is 1, to line 20 if the value of VAR is 2, etc. In general control is transferred to the Nth line number in the list, with the value of N determined by the given formula. Try the following program, which uses the ON / GOTO command to type a message appropriate to the number you supply.