CHAPTER 10
Odds and Ends

Pythagorean triplets

Recall the theorem of Pythagoras which has already been mentioned in this book. It states that in a right-angled triangle with sides X, Y and Z, where Z is the hypotenuse (the longest side), the following relation holds:

$$X^2 + Y^2 = Z^2$$

The classic example that is readily recalled is the 3, 4, 5 right triangle.

$$3^2 + 4^2 = 5^2$$

Another example is the 5, 12, 13 right triangle.

$$5^2 + 12^2 = 13^2$$

Of course there are infinitely many different examples of right-angled triangles. But, how many examples are there in which X, Y and Z are integers? The answer is that there are infinitely many. Such numbers X, Y and Z are called *Pythagorean triplets*.

How can we produce a list of Pythagorean triplets? Naturally such a list should not contain triplets that are products of another. For instance, since 3, 4, 5 is a Pythagorean triplet so is 6, 8, 10. Pythagorean triplets which have no common factor are called *primitive*. Thus 3, 4, 5 is a primitive Pythagorean triplet while 6, 8, 10 is not primitive.

Producing Pythagorean triplets could be time-consuming were it not for some mathematicians who managed to produce an elegant method of generating primitive Pythagorean triplets. The technique is as follows.

1. Choose two positive integers A and B so that:

    (a) A is greater than B.
    (b) A + B is odd (thus one of the numbers is odd and the other is even).
    (c) A and B have no common divisor except 1.

2. Calculate

   X = A2 - B2
   Y = 2*A*B
   Z = A2 + B2

The numbers X, Y, Z are a primitive Pythagorean triplet. Conversely, any primitive Pythagorean triplet can be formed in this way. This technique is used in the program Pythagorean Triplets to produce a list of Pythagorean triplets. The process starts with A = 2. The program then finds all possible values of B satisfying the required conditions. The value of A is increased and the process repeated. Twenty triplets are printed out at a time. You may continue for as long as you like. The starting value of A may be changed if desired.

Listing 10.1

LIST
```
   10 REM Pythagorea triplets
   20 MODE 1:COLOUR 3:PRINT ' TAB(10);"P
ythagorean triplets"':@%=9:COLOUR 1
   30 PRINT "This program prints out som
e primitive  Pythagorean triplets."'
   40 PRINT ,"  Press Y to start"
   50 K%=0:A%=2:B%=3
   60 REPEAT:G$=GET$:UNTIL G$="Y"
   70 REPEAT
   80  CLS:COLOUR 3:PRINT ' TAB(10);"Pyt
hagorean triplets"'
   90  PRINT "      Count    * X *     * Y
 *    * Z *":COLOUR 1
  100  REPEAT
  110   B%=B%-2:IF B%<1 THEN A%=A%+1:B%=
A%-1
  120   X%=A%:Y%=B%
  130   REPEAT
  140    N%=X% DIV Y%:Z%=X%-N%*Y%
  150    IF Z%<>0 THEN X%=Y%:Y%=Z%
  160   UNTIL Z%=0
  170   IF Y%<>1 THEN TEST=0 ELSE TEST=-
1
```

```
  180   IF TEST:K%=K%+1:PRINT K% A%*A%-B
%*B% 2*A%*B% A%*A%+B%*B%
  190   TEST=TEST*(K% MOD 20 = 0)
  200  UNTIL TEST
  210  COLOUR 2:PRINT ' TAB(11);"Continu
e? Y or N ";
  220  REPEAT:G$=GET$:UNTIL G$="Y" OR G$
="N"
  230 UNTIL G$="N"
  240 COLOUR 3:PRINT CHR$(7) '' TAB(10);
"Another go? Y or N ";
  250 REPEAT:G$=GET$:UNTIL G$="Y" OR G$=
"N"
  260 IF G$="Y" THEN RUN
  270 CLS:PRINT '"Bye for now.":@%=10:EN
D
```

RUN

```
          Pythagorean triplets

      Count    * X *     * Y *     * Z *
        1         3         4         5
        2         5        12        13
        3         7        24        25
        4        15         8        17
        5         9        40        41
        6        21        20        29
        7        11        60        61
        8        35        12        37
        9        13        84        85
       10        33        56        65
       11        45        28        53
       12        15       112       113
       13        39        80        89
       14        55        48        73
       15        63        16        65
       16        17       144       145
       17        65        72        97
       18        77        36        85
```

```
 19         19        180        181
 20         51        140        149

     Continue? Y or N



    Pythagorean triplets

Count    * X *     * Y *     * Z *
 21         91         60        109
 22         99         20        101
 23         21        220        221
 24         57        176        185
 25         85        132        157
 26        105         88        137
 27        117         44        125
 28         23        264        265
 29         95        168        193
 30        119        120        169
 31        143         24        145
 32         25        312        313
 33         69        260        269
 34        105        208        233
 35        133        156        205
 36        153        104        185
 37        165         52        173
 38         27        364        365
 39         75        308        317
 40        115        252        277

     Continue? Y or N
```

Multi-precision powers

The BBC and Electron keep numbers with 9 significant digits accurately. Large numbers are rounded off. How can we calculate large powers of numbers accurately? For instance, what is 2 to the power of 130? The answer is to use multi-precision arithmetic.

One way to produce multi-precision arithmetic is to store the digits of a number in an array M(I), with M(0) storing the last 4 digits, M(1) storing the previous 4 and so on. Thus 987654 would be stored as:

M(1) = 98, M(0) = 7654

The original number is recovered by converting the array elements into strings and printing each in turn. Of course, it may also be recovered by the formula:

M = M(1)*104 + M(0)

Suppose that we want to multiply M = 987654 by N = 23456. First we would write these numbers into arrays M(I) and N(I).

M(1) = 98, M(0) = 7654
N(1) = 2, N(0) = 3456

Now form the following

C(0) = M(0)*N(0)
C(1) = M(1)*N(0) + M(0)*N(1)
C(2) = M(1)*N(1)

The result becomes:

C(0) = 26452224
C(1) = 353996
C(2) = 196

We then calculate the product M*N by the following

196*108 + 353996*104 + 26452224
= 19600000000 + 3539960000 + 26452224
= 23166412224

Of course the calculation is not done as shown above - your computer would simply round off the numbers. What we do is strip off the digits from the left of C(0) so that only 4 are left. We add the digits stripped off to C(1).

C(0) $\rightarrow$ 2224
C(1) $\rightarrow$ 353996 + 2645 = 356641

Next, strip off the digits from the left of C(1) so that only 4 are left. Add the digits stipped off to C(2).

C(1) $\rightarrow$ 6641
C(2) $\rightarrow$ 196 + 35 = 231

We therefore obtain the following:

C(2) = 231, C(1) = 6641, C(0) = 2224

from which we read immediately that

987654*23456 = 23166412224

This process is quite general. For other larger numbers there may be a value for the array element M(2), M(3), etc. In such a situation we would form the following.

C(0) = M(0)*N(0)
C(1) = M(1)*N(0) + M(0)*N(1)
C(2) = M(2)*N(0) + M(1)*N(1) + M(0)*N(2)
etc.

After this the stripping process is performed and the answer can finally be printed out.

The program Multi-Precision Powers illustrates how the process just described may be used to calculate accurately powers of numbers. The program continues until a 40 digit number is reached. If desired you can have even higher degrees of accuracy by changing the value of X% in line 90. The value of X% + 1 times 4 is the degree of precision.

Listing 10.2
LIST
```
   10 REM Multi-precision powers
   20 MODE 1:COLOUR 3:PRINT ' TAB(9);"Mu
lti-precision powers"'
   30 PRINT "This program prints out pow
ers of an    integer accurate to 40 digi
ts."'
   40 COLOUR 1:PRINT "Enter number whose
 powers are required."
   50 REPEAT
   60  INPUT '"Number? " N%
   70  IF N%<2 OR N%>999999999 THEN COLO
UR 3:PRINT'"Try a sensible number.":COLO
UR 1
   80 UNTIL N%>1 AND N%<1E10
   90 T%=10000:X%=9:Y%=X%+2:DIM M%(X%),N
```

```
%(X%),L%(Y%):K%=1:N%(0)=N%:M%(0)=N%
  100 REM Change X above for other degre
es of precision
  110 FOR I%=0 TO 2
  120  IF M%(I%)>=T% THEN M%(I%+1)=M%(I%
+1)+(M%(I%) DIV T%):M%(I%)=M%(I%) MOD T%
  130  N%(I%)=M%(I%)
  140 NEXT
  150 REPEAT
  160  CLS:PRINT '"Multi-precision power
s of ";N% '
  170  REPEAT
  180   K%=K%+1
  190   FOR I%=0 TO Y%:L%(I%)=0:NEXT
  200   FOR J%=0 TO 2:FOR I%=0 TO X%:L%(
I%+J%)=L%(I%+J%)+N%(I%)*M%(J%):NEXT:NEXT
  210   FOR I%=0 TO X%:N%(I%)=L%(I%):NEX
T
  220   FOR I%=0 TO X%
  230    IF N%(I%)>=T% THEN N%(I%+1)=N%(
I%+1)+(N%(I%) DIV T%):N%(I%)=N%(I%) MOD
T%
  240   NEXT
  250   FOR I%=0 TO X%
  260    IF N%(I%)>0 THEN L%=I%
  270   NEXT
  280   COLOUR 1:PRINT ;N%;"^";K%;"=":CO
LOUR 2
  290   FOR I%=L% TO 0 STEP -1
  300    A$=STR$(N%(I%))
  310    REPEAT
  320     IF LEN(A$)<>4 THEN A$="0"+A$
  330    UNTIL LEN(A$)=4
  340    IF I%=L% THEN PRINT ;VAL(A$); E
LSE PRINT A$;
  350   NEXT
  360   PRINT
  370  UNTIL K% MOD 10 = 0 OR (N%(X%)*T%
```

```
+N%(X%-1))*N%>=T%*T%
  380  COLOUR 3:PRINT CHR$(7) ' TAB(11);
"Press Y to continue. ";
  390  REPEAT:G$=GET$:UNTIL G$="Y"
  400 UNTIL (N%(X%)*T%+N%(X%-1))*N%>=T%*
T%
  410 COLOUR 3:PRINT CHR$(7) '' TAB(10);
"Another go? Y or N "
  420 REPEAT:G$=GET$:UNTIL G$="Y" OR G$=
"N"
  430 IF G$="Y" THEN RUN
  440 CLS:PRINT '"Bye for now.":END
```

RUN

```
         Multi-precision powers

This program prints out powers of an
integer accurate to 40 digits.

Enter number whose powers are required.

Number? 128


Multi-precision powers of 128

128^2=
16384
128^3=
2097152
128^4=
268435456
128^5=
34359738368
128^6=
4398046511104
128^7=
```

```
562949953421312
128^8=
72057594037927936
128^9=
9223372036854775808
128^10=
1180591620717411303424

           Press Y to continue.

Multi-precision powers of 128

128^11=
151115727451828646838272
128^12=
19342813113834066795298816
128^13=
2475880078570760549798248448
128^14=
316912650057057350374175801344
128^15=
40564819207303340847894502572032
128^16=
5192296858534827628530496329220096
128^17=
664613997892457936451903530140172288
128^18=
85070591730234615865843651857942052864

           Press Y to continue.

          Another go? Y or N
```