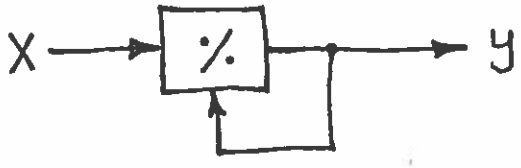
Home-Canned Pickles, Preserves, And SQUARE ROOTS

A Problem You'll Relish!

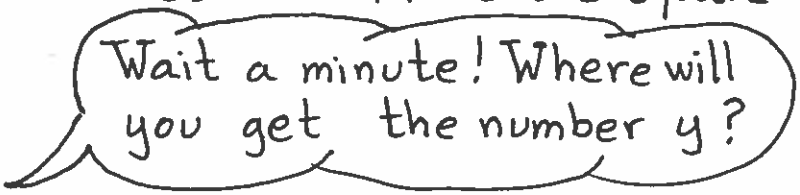
Suppose, just to live life the hard way, you wanted to write your own computer program to calculate square roots. You might start by observing that if

$$y = \sqrt{x}, \text{ then } y^2 = x \text{ or } y = \frac{x}{y}$$

This last equation gives you a brilliant idea. To find the square root of x, you simply need to divide x by its square root. The result is $\sqrt{x}$! Joyfully you picture this in your mind:



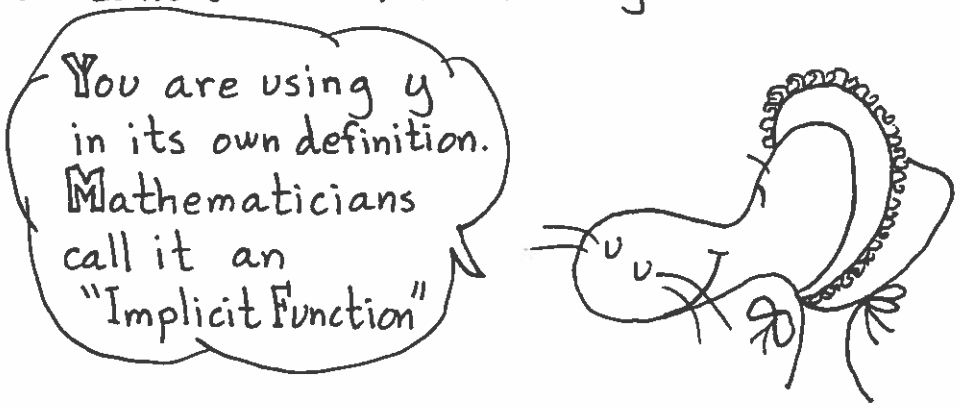
All you need is a machine that will divide a number x by a number y. The number that comes out will be the square root of x!



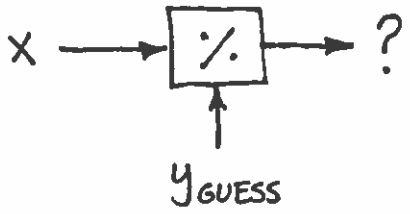
Out of the machine! We'll call that y!

Suddenly you are chicken. You need to divide x by y in order to get the answer. But y is the answer, so you need the answer 1st to divide into x, but you are egged on by why y comes out before it went in?? Why, oh why, O y?

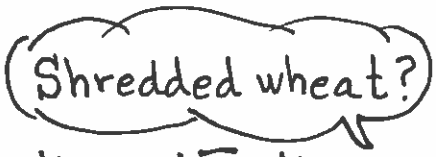
Bolzano's bambino tells you



Illicit you could relate to, but implicit doesn't help you at all. But you get an idea! Suppose I guess a value of y and feed it into my box.



What will come out?



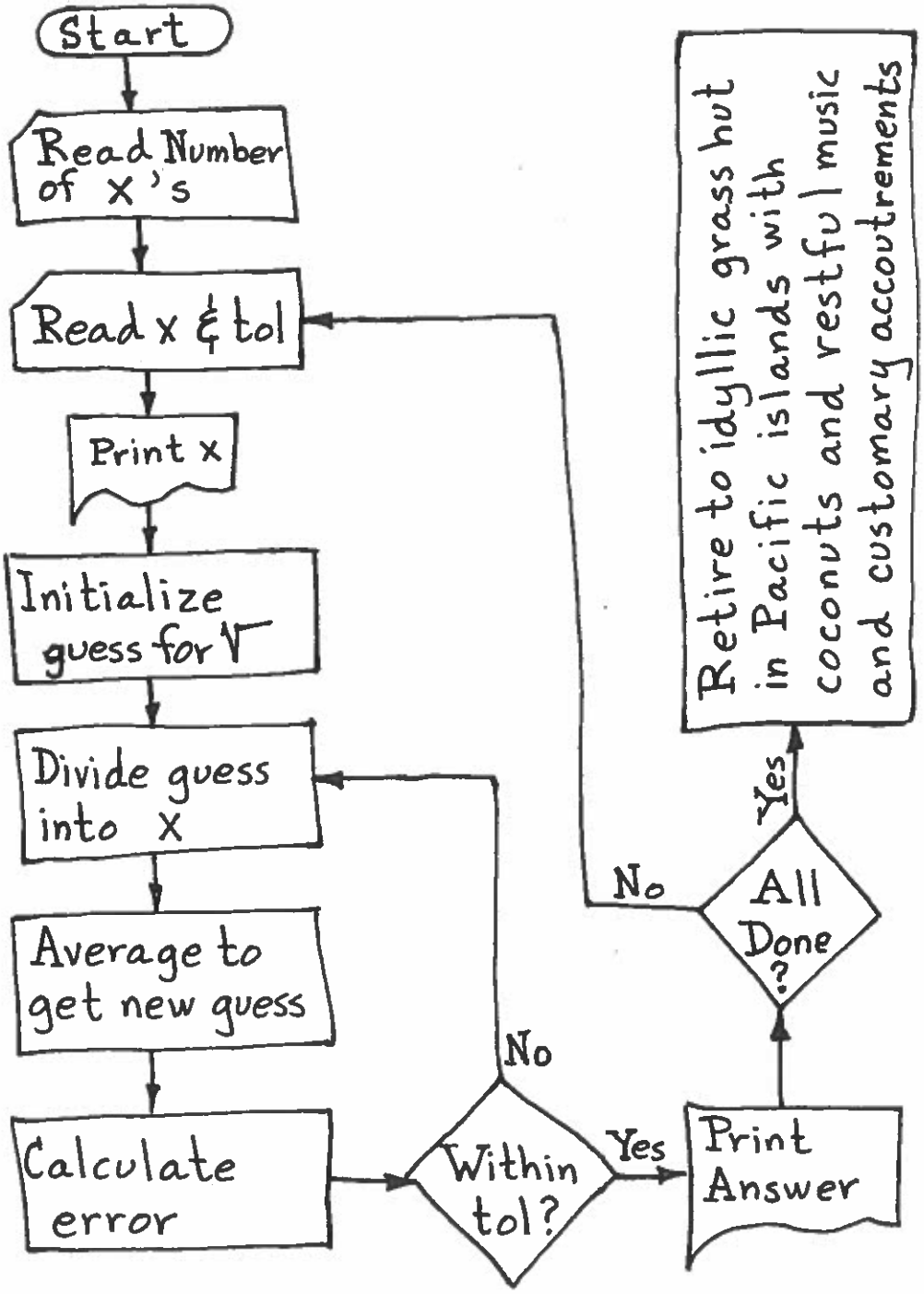
No! If the guess is less than $\sqrt{x}$, then what comes out will be more! And if the guess is too big, what comes out will be too small.

So? So it's all sewn up! Whatever I guess, I know the true $\sqrt{x}$ lies between my guess and whatever comes out! These two numbers are upper and lower bounds on the true $\sqrt{x}$. So? So if I average those two numbers, I'll have an even better guess. I can try again and again till I'm as close as I want to get!

Wait a minute! It would take you forever to find the exact root

And the computer only keeps about six figures of accuracy in its calculations, so even if you let it run forever, it still might not find the exact answer!

So who needs to be that precise? I'll stop when I'm within a reasonable tolerance of the true answer. Here's what I'll do:


Start
Read Number of X's
Read x & tol
Print x
Initialize guess for √
Divide guess into X
Average to get new guess
Calculate error
Within tol?
No
Yes
Print Answer
All Done ..?
No
Yes
Retire to idyllic grass hut in Pacific islands with coconuts and restful music and customary accoutrements

And here's my actual program:

```
//bJOB   DR. K  } These control cards will be
//bFOR          } different on your computer!
C SQUARE ROOT SAMPLE PROGRAM
C READ NUMBER OF CARDS WITH ROOTS TO FIND
      READ(8,10)N
10    FORMAT(I5)
C ECHO PRINT THE INPUT DATA
      WRITE(5,20) N
20    FORMAT('1THE NUMBER OF ROOTS IS',I5///)
C READ NUMBER WHOSE ROOT IS DESIRED
C AND TOLERANCE REQUIRED
30    READ(8,40) X, TOL
40    FORMAT( 2F10.4)
C ECHO PRINT TO BE SAFE
      WRITE(5,50) X, TOL
50    FORMAT('bX=b',F10.4,5X,
     1'bTOLERANCE ISb',F10.4)
C INITIALIZE Y GUESS
      YG = X
60    TEMP= X / YG
CALCULATE NEW GUESS FOR ROOT
      YG = 0.5*( YG + TEMP)
CALCULATE THE ERROR-CHECK AGAINST X IN
CASE OUR THEORY ITSELF HAS A BUG
      ERROR = ABS (X - YG * YG)
```

```
C LET'S WATCH THE ALGORITHM CONVERGE
      WRITE(5,70) ERROR
70    FORMAT(15X,'ERROR IS b',E12.5)
C LOOP BACK IF NOT WITHIN TOLERANCE
      IF(ERROR.GT.TOL) GO TO 60
C PRINT FINAL ANSWER
      WRITE(5,80) X,YG
80    FORMAT(/'bTHE SQUARE ROOT OF b',
     1F10.4,'bIS b',F15.6//)
C ARE THERE MORE ROOTS TO FIND?
      N=N-1
      IF(N.GT.0) GO TO 30
      WRITE(5,90)
90    FORMAT(///'bTHTHTHAT"S ALL,b
     1FOLKS')
      CALL EXIT
      END
//bXEQ
bbbb4
  2.
   17.62
 25.
8.36
//bEND
```

```
0.001
 0.0005
0.0001
     0.001
```

Columns 11-20

These cards may differ on your computer system, so check!