Ponder this

http://www.research.ibm.com/haifa/ponderthis/

Ponder this (January 2004)

- This month's puzzle was sent in by Joe Buhler. It came from a SIGCSE meeting via Eric Roberts.
- A read-only array of length n, with address from 1 to n inclusive, contains entries from the set {1, 2, ..., n-1}. By Dirichlet's Pigeon-Hole Principle there are one or more duplicated entries.
- Find a linear-time algorithm that prints a duplicated value, using only "constant extra space". (This space restriction is important; we have only a fixed number of usable read/write memory locations, each capable of storing an integer between 1 and n. The number of such locations is constant, independent of n. The original array entries can not be altered.)
- The algorithm should be easily implementable in any standard programming language.

Solution

- This technique is known as Pollard's rho method; it was developed by John Pollard as a tool for integer factorization.
- Let f(x) be the location of the array at address x.

```
a := f(f(n))
b := f(n)
do while not (a=b)
a:=f(f(a))
b:=f(b)
end
/* Now a=b can be reached at either 2k or k steps from n, */
/* where k is some integer between 1 and n. */
a:=n
do while not (a=b)
a:=f(a)
b:=f(b)
end
print a
```

Solution

- The pattern that you get, starting from n and repeatedly applying f (doing the table lookup), is a string of some length $L>0$ leading to a cycle of some length M, with $L+M<n$. (L is nonzero because n is outside the range of f.)

- It is shaped like the Greek letter rho, hence its name. Our stopping count k is the least multiple of M greater than or equal to L, so that $L \leq k < L+M < n$.