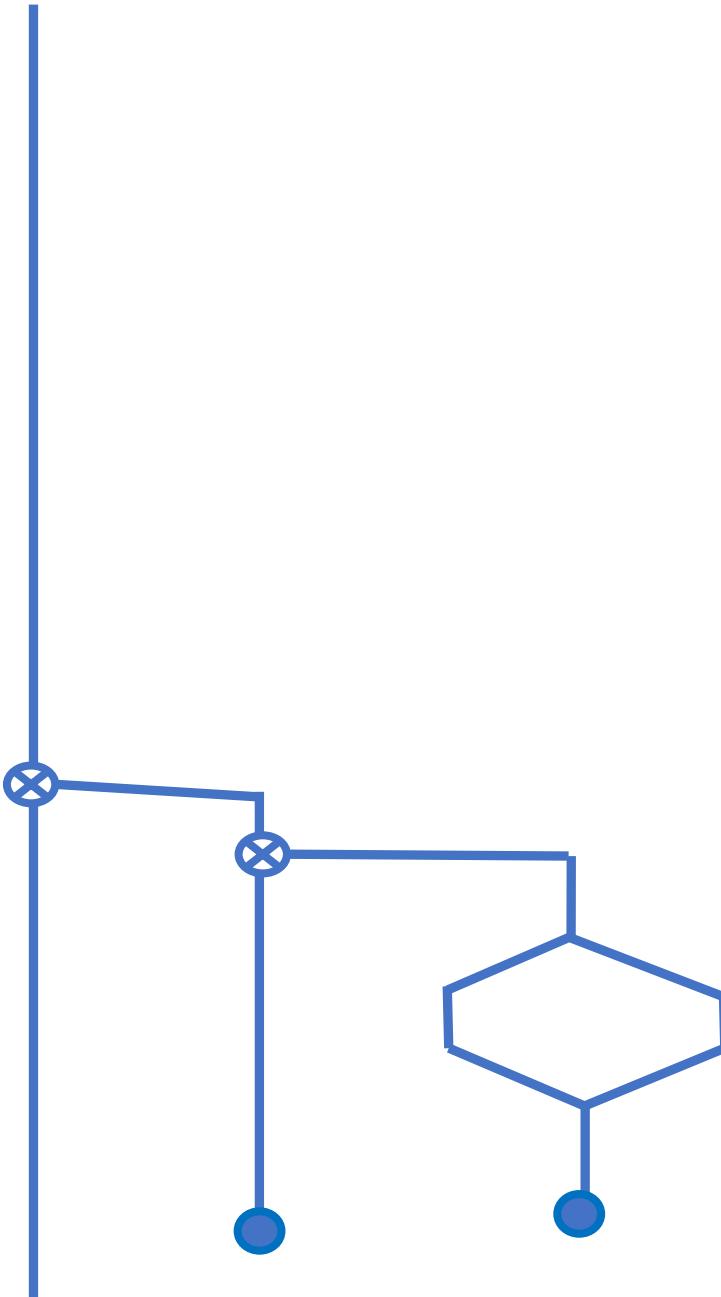


Challenge: A Poor Man's Floating Point

SVFIG Zoom
Dec. 18, 2021
Bill Ragsdale



The Problem

- ❖ Compute (approximately) $100!$
- ❖ Check with floating point math.
- ❖ Calculate the number of trailing zeros.
- ❖ Extra Credit: Compute $100!$ exactly.

Pseudocode

Setup variables

Setup a loop from 2 to $N+1$.

Initial value of 1 (one) as the product.

Within the loop

Form a new product by multiplying the product by the loop index.

Adjust for overflow.

Loop to completion

Report

The Setup

variable	Significand	\ significand of result
variable	Exponent	\ exponent of result
variable	Input	\ Number for factorial

The Setup

variable **Significand** \ significand of result

variable **Exponent** \ exponent of result

variable **Input** \ Number for factorial

```
: setup 100 Input      ! \ For the calculation
        0  Significand !
        0  Exponent   ! ;
```

The Setup

```
variable Significand \ significand of result
variable Exponent   \ exponent of result
variable Input     \ Number for factorial

: setup 100 Input      ! \ For the calculation
        0  Significand !
        0  Exponent    ! ;

: .output \ convert for printing
  <# Exponent @ s>d #s 2drop
  ascii e hold
  Significand @ s>d #s #> type ;
```

The Wrapper

```
: by-integer      setup  process .output ;

: process          \ process for a factorial
  1 Input @ 1+ 2   \ loop values
do  i adjust loop
drop ;
```

The Action

```
: adjust ( multiplicand i --- product )  
begin  
  2dup um*          \ a double cell product
```


The Action

```
: adjust ( multiplicand i --- product )
begin
  2dup um*          \ a double cell product
while              \ test for high cell overflow
  drop swap        \ get the product
  0 10 UM/MOD      \ divide product by ten
  nip swap         \
  1 Exponent +!   \ increment Exponent by 1
repeat
```

The Action

```
: adjust ( multiplicand i --- product )
begin
  2dup um*          \ a double cell product
while              \ test for high cell overflow
  drop swap        \ get the product
  0 10 UM/MOD      \ divide product by ten
  nip swap         \
  1 Exponent +!   \ increment Exponent by 1
repeat
  nip nip          \ trim and save Significant
  dup Significant ? ;
```

The Check

By the Win32Forth floating point.

```
: by-floats
  1e0
  Input @ 1+ 2    do i s>f f* loop
  FE. ;
```

A Test

```
setup process .output
```

```
933260130e149 ok (use better rounding)
```

```
by-floats
```

```
93.326215E156 ok
```

9.332622e+157

Rad	Deg	x!	()	%	AC
Inv	sin	ln	7	8	9	÷
π	cos	log	4	5	6	×
e	tan	√	1	2	3	-
Ans	EXP	x ^y	0	.	=	+

Exact Solution, by Wolfram

9332621544394415268169923885626670
0490715968264381621468592963895217
5999932299156089414639761565182862
53697920827223758251185210916864
000000000000000000000000000000

How Many Zeros?

Sum the number of fives as factors.

$$100 / 5 = 20$$

$$20 / 5 = \underline{+ 4}$$

24 zeros

9332621544394415268169923885626670

0490715968264381621468592963895217

5999932299156089414639761565182862

53697920827223758251185210916864

00000 00000 00000 00000 0000

Zeros In Forth

```
: zeros?  
  0 swap  
  begin  5 / dup >r + r>  \ result quotient  
    dup 5 <              \ res quot limit  
  until  
    drop  
cr cr ." Number of trailing zeros is " . ;
```

```
100 zeros?
```

```
Number of trailing zeros is 24 ok
```

Discoveries

My LOOP was 100 2 DO ... LOOP which ran 2 to 99. Off by a factor of 100. 1+ fixes.

My value is low due to always rounding down after division.

Better to round down if the least significant digit is odd and up if it is even.

Credits

- Google and Wolfram.com for revealing my DO LOOP error.
- Andrew McKewan and Tom Zimmer for Win32Forth.
- And the Europeans for their updates.