

NAME

bigfact – Approximate factorials of large integers.

SYNOPSIS

`${HOME}/bin/exe/bigfact 123`

DESCRIPTION

If the program is run with no command-line parameter it prints usage information and stops with return code 1. Otherwise it obtains from its single parameter the integer N whose factorial is required. If the parameter is not an integer, or if it is negative, the program prints an error message and stops with return code 2. If N does not exceed 12, the program prints its exact factorial from a table of values. If N exceeds 12 but does not exceed 170, the program prints its REAL*8 factorial from a table of values.

For $N > 170$ the program approximates $N!$ as follows. First it computes $\log(N!) = \log(1) + \log(2) + \dots + \log(N)$. If this quantity exceeds 2147483647 the program prints an error message and resigns with return code 3. Otherwise it uses truncation and subtraction to express $\log(N!)$ as $10^A \times 10^P = F \times 10^P$, where P is an integer and $F = 10^A$ is a REAL*8 number in the interval [1,10).

When the program runs without exception it stops with return code 0. Values of $N!$ reported for $N < 13$ are integer and thus exact. Values of $N!$ for N from 14 to 21 have few enough significant digits that they could in principle be reported exactly by this program, but because their decimal values are stored as floating-point numbers the last digit printed differs from that tabled. Numbers N larger than 21 have more significant digits than can be printed by this program, so the reported values are necessarily approximate. However, the factorials reported for $N=1000$, $N=10000$, and $N=100000$ agree with those given in <https://en.wikipedia.org/wiki/Factorial> through the digits given there, and the factorial reported for $N=1000000$ agrees with that table in its first 6 significant digits.

UNITS and FILES

0 error messages
6 output

DIAGNOSTICS

The possible return codes are summarized in the table below.

0 all went well
1 no parameter was given
2 N is not a nonnegative integer
3 N is too big
4 the conversion of P to numerals failed

AUTHOR

Michael Kupferschmid

EXAMPLE

```
unix[1] bigfact
usage: bigfact N
       returns factorial of nonnegative integer N
unix[2] bigfact -1
N must be nonnegative but it is      -1
unix[3] bigfact 0
1
unix[4] bigfact 12
479001600
unix[5] bigfact 13
6.227020800000002 x 10^9
unix[6] bigfact 1000
4.023872600748676 x 10^2567
unix[7] bigfact 1000000
8.263930498920278 x 10^5565708
unix[8] bigfact 1000000000
log10 N!= 8.565705522950354D+09 too big for this algorithm
unix[9]
```