Re: Number to Letter Counting algorithim needed

Source: http://sci.tech−archive.net/Archive/sci.math/2007−06/msg03621.html

- *From*: "Wlodzimierz Holsztynski (wlod)" <sennajawa@xxxxxxxxx>

- *Date*: Tue, 19 Jun 2007 20:09:09 −0000

On Jun 19, 10:50 am, jgam...@xxxxxxxxx (John M. Gamble) wrote:

> In article <46743c54$0$8716$ed261...@xxxxxxxxxxxxxxxxxxxxxxxxxxx>,

> a) You realize that comp.lang.perl.misc
> is the correct place to post
> this sort of question, right?

Nonsense. The question had nothing to do specifically with perl. It has a simple mathematical element to it, which is of mathematical interest. After all, the positional representation of natural numbers is of a fundamental importance to mathematics. Thus it's also important to understand their variations.

At the end of this post I will explain that the given version of encoding natural numbers is clearly more efficient w.r. to compression than the standard positional way (no, not really "better" :−).

For instance, Bill in this thread didn't have a clue. We are dealing here still with base 26, and not with base 27. The question was simple but not everybody was able to understand the issue.

> b) This sort of incrementing can be handled
> quite naturally in perl.

Why perl? Most any language, starting with
good ol' C, will be more than adequate.
Don't get offended, but there is an old,
great saying:

"you're shitting above your asshole",

meaning in this case that you and Bill are
talking about things which are above your
present understanding.

> c) If there are side issues that affect b),
> then either CPAN will have somthing to handle
> it, or you can ask for help on
> comp.lang.perl.misc.

Perhaps perl groups would be ok too.
But it was perfectly proper to ask
here, at sci.math. The question is
first of all of a mathematical interest,
and only then of an interest to computer
science (informatics :−), in particular
to the theory of data compression.
All this within its miniature scale.

***

Lets have just two digits: 0 and 1.
We would like to send an ordered list
of natural numbers, separated by blank.
The "excell" codes are a bit shorter
than the standard positional codes:

1 −− 0
10 −− 1
11 −− 00
100 −− 01
101 −− 10
110 −− 11
111 −− 000
1000 −− 001
1001 −− 010

etc.

The excell encoding will produce lists
which are never longer, and often shorter
than the positionally encoded lists.

Regards,

Wlod

.