2012-01-16

Combien de zéros à la fin de 1000! ?

Par définition, $\forall n \in \mathbb{N}^*$, $n! = 1\times 2\times 3\times \cdots \times n = \prod_{k=1}^{n} k$.

Avant de nous attaquer à 1000!, regardons ce qu'il se passe pour 30!. Nous donnons ci-dessous pour chaque nombre k de $\prod_{k=1}^{n} k$ ses facteurs premiers.

$1 \rightsquigarrow$ Pas de facteurs premiers	$11 \rightsquigarrow 11$	$21 \rightsquigarrow 3\times 7$
$2 \rightsquigarrow 2$ lui-même	$12 \rightsquigarrow 2^2\times 3$	$22 \rightsquigarrow 2\times 11$
$3 \rightsquigarrow 3$	$13 \rightsquigarrow 13$	$23 \rightsquigarrow 23$
$4 \rightsquigarrow 2\times 2 = 2^2$	$14 \rightsquigarrow 2\times 7$	$24 \rightsquigarrow 2^3\times 3$
$5 \rightsquigarrow 5$	$15 \rightsquigarrow 3\times 5$	$25 \rightsquigarrow 5^2$
$6 \rightsquigarrow 2\times 3$	$16 \rightsquigarrow 2^4$	$26 \rightsquigarrow 2\times 13$
$7 \rightsquigarrow 7$	$17 \rightsquigarrow 17$	$27 \rightsquigarrow 3^3$
$8 \rightsquigarrow 2^3$	$18 \rightsquigarrow 2\times 3^2$	$28 \rightsquigarrow 2^2\times 7$
$9 \rightsquigarrow 3^2$	$19 \rightsquigarrow 19$	$29 \rightsquigarrow 29$
$10 \rightsquigarrow 2\times 5$	$20 \rightsquigarrow 2^2\times 5$	$30 \rightsquigarrow 2\times 3\times 5$

Combien "fabrique-t-on" des zéros à la fin de 30! ? Il faut partir à la recherche des 10 que l'on peut former à l'aide des facteurs premiers donnés ci-dessus.

car $10 = 5\times 2$

Commençons par chercher les cinq. Ils apparaissent dans 5, 10, 15, 20, 25 et 30, autrement dit dans les multiples de 5 compris entre 1 et 30 au sens large. Cela nous fournit $6+1 = 7$ facteurs premiers égaux à 5 dans notre tableau ci-dessus (le +1 vient de ce que $25 = 5^2$).

Pour obtenir un 10, chaque 5 doit être multiplié par un 2.

A-t-on assez de 2 ? Clairement oui ! Par exemple, il y a $\frac{30}{2} = 15$ multiples de 2 appartenant à $\{1; 2; \cdots; 30\}$.

Nous avons maintenant tous les ingrédients pour répondre à notre question hautement métaphysique.

• Cherchons les 5 apparaissant dans $\prod_{k=1}^{1000} k$. Il y a $\frac{1000}{5} = 200$ multiples de 5 dans $\{1; 2; \cdots; 1000\}$. Chacun d'eux va fournir un 5 au moins. En fait, il faut ensuite considérer les multiples de $25 = 5^2$: chacun d'eux va fournir un 5 de plus. Il y a $\frac{1000}{25} = 40$ multiples de 25. Ensuite, les multiples de $125 = 5^3$ vont fournir $\frac{1000}{125} = 8$ facteurs premiers égaux à 5 de plus. Enfin, comme $5^4 = 625$ et $\frac{1000}{625} = 1,6$, nous avons juste 625 comme multiple de 625 dans $[1; 1000]$ (bien entendu, on peut voir cela directement).

Au total, cela nous fait :

$$\underbrace{200} + \underbrace{40} + \underbrace{8} + \underbrace{1} = 249$$

Un 5 fourni par chaque multiple de 5.

Un 5 de plus fourni par chaque multiple de 25 qui est aussi un multiple de 5.

Un 5 de plus fourni par chaque multiple de 125 qui est aussi un multiple de 25 et 5.

Un 5 de plus fourni par 625 qui est aussi multiple de 5, 25 et 125.

• A-t-on assez de 2 ? Oui car il y a $\frac{1000}{2} = 500$ multiples de 2 dans $[1; 1000]$.

En résumé, 1000! se finit par 249 zéros.

Finissons avec une solution de passionné informatique. Ce qui suit a été fait à l'aide de SAGE (voir sagemath sur Google).

sage The Sage Notebook
Version 4.7.2

Christophe_BAL Toggle | Home | Published | Log | Settings | Help | Report a Problem | Sign out

Factoriel_1000
last edited on Jan 16, 2012 3:30:45 PM by Christophe_BAL

Save | Save & quit | Discard & quit

File... | Action... | Data... | sage | ☐ Typeset   Print Worksheet Edit Text Revisions Share Publish

```
print factorial(1000)
```

```
402387260077093773543702433923003985719374864210714632543799910429938512\
398629020592044208486969404800479988610197196058631666872994808558901323\
829669944590997424504087073759918823627727188732519779505950995276120874\
975462497043601418278094646496291056393887437886487337119181045825783647\
849977012476632889835955735432513185323958463075557409114262417474349347\
553428646576611667797396668820291207379143853719588249808126867838374559\
731746136085379534524221586593201928090878297308431392844403281231558611\
036976801357304216168747609675871348312025478589320767169132448426236131\
412508780208000261683151027341827977704784635868170164365024153691398281\
264810213092761244896359928705114964975419909342221566832572080821333186\
116811553615836546984046708975602900950537616475847728421889679646244945\
160765353408198901385442487984959953319101723355556602139450399736280750\
137837615307127761926849034352625200015888535147331611702103968175921510\
907788019393178114194545257223865541461062892187960223838971476088506276\
862967146674697562911234082439208160153780889893964518263243671616762179\
168909779911903754031274622289988005195444414282012187361745992642956581\
746628302955570299024324153181617210465832036786906117260158783520751516\
284225540265170483304226143974286933061690897968482590125458327168226458\
066526769958652682272807075781391858178889652208164348344825993266043367\
660176999612831860788386150279465955131156552036093988180612138558600301\
435694527224206344631797460594682573103790084024432438465657245014402821\
885252470935190620929023136493273497565513958720559654228749774011413346\
962715422845862377387538230483865688976461927383814900140767310446640259\
899490222221765904339901886018566526485061799702356193897017860040811889\
729918311021171229845901641921068884387121855646124960798722908519296819\
372388642614839657382291123125024186649353143970137428531926649875337218\
940694281434118520158014123344828015051399694290153483077644569099073152\
433278288269864602789864321139083506217095002597389863554277196742822248\
757586765752344220207573630569498825087968928162753848863396909959826280\
956121450994871701244516461260379029309120889086942028510640182154399457\
156805941872748998094254742173582401063677404595741785160829230135358081\
840096996372524230560855903700624271243416909004153690105933983835777939\
410970027753472000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000
```

```
print len("000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000")
```

```
249
```

evaluate