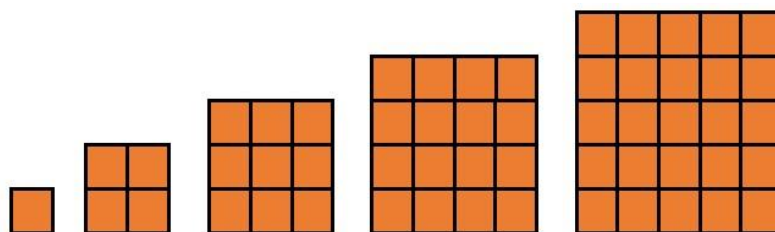


2018 Novembro

Problema 2: Resolução



Esta foi a explicação que o Pedro deixou para te transmitir.

Se tiveres um inteiro positivo ele pode escrever-se, de forma única, como produto dos seus factores primos¹:

$$N = p_1^{m_1} \times p_2^{m_2} \times \dots \times p_k^{m_k}$$

Se substituíres quaisquer dos expoentes m_i por um inteiro entre 0 e m_i obténs, obviamente, um divisor de N .

E, claro, qualquer divisor de N não pode ter outros divisores primos para além dos p 's nem os seus expoentes podem ser superiores aos m_i 's logo desta forma podes obter todos os divisores de N .

O número de divisores de N é, por isso, o número de formas diferentes de colocar nos expoentes de cada p_i um número inteiro entre 0 e m_i , ou seja:

$$\Pi = (m_1 + 1) \times (m_2 + 1) \dots \times (m_k + 1)$$

Repara agora que a rotina do programa altera o valor da posição P nas vezes em que o valor testado é um divisor de P maior ou igual a dois. Para que a posição mantenha, no fim, o valor inicial o número de alterações, igual ao número de divisores maiores ou iguais a dois, deve ser par.

Mas então o número de divisores, Π , deve ser ímpar, pois um também é divisor, o que acontece se e só se os m_i 's forem todos pares ou seja se e só se P for um quadrado perfeito.

Assim as posições que mantêm o valor são os quadrados perfeitos entre 1 e 100, 1 inclusive pois nunca altera o valor.

¹ Teorema Fundamental da Álgebra

O Pedro resolveu escrever este programa de meia dúzia de linhas, em Just Basic, para confirmar o seu raciocínio.

E, lembrando-se de ti, acrescentou alguns comentários (precedidos de rem no texto) para que o possas entender facilmente.

```
rem reserva 100 posições, N(1), N(2),...,N(100), para guardar 100 números
Dim N(100)

rem coloca 1 em todas as posições: N(i)= 1 para i de 1 a 100
for i=1 to 100
N(i)=1
next i

rem para cada i de 2 a 100 vai alterar, de 0 para 1 e de 1 para 0, o valor das posições j onde j é múltiplo de i
for i=2 to 100
for j=i to 100

rem testa se j é múltiplo de i
if j mod i = 0 then

rem se for altera o valor N(j)
if N(j)=0 then
N(j)=1
else
N(j)=0
end if
end if

next j
next i

rem imprime as posições que guardam o valor 1
for i=1 to 100

if N(i) = 1 then
Print i
end if

next i
```

Agora podes fazer o download do Just Basic V1.01, [aqui](#), introduzir estas linhas no editor e corrê-lo para veres o resultado.

E substituir neste programa o 100 por qualquer outro número e ver o que obténs... para te divertires!

Sabes que o Pedro resolveu substituir o 100 por 16000?... fartou-se de secar à espera do resultado!

Até que... teve uma ideia!



Escreveu meia dúzia de linhas no editor e obteve um programa que produzia o mesmo output, mas muito rapidamente.

Queres tentar?