

Code examples

Haskell

```

1 import Data.Ratio ( (%) )
2 import GHC.Real ( Ratio((:)) )
3 import Data.Function ( on )
4 import Data.List ( intercalate )
5 import GHC.Float.RealFracMethods ( truncateDoubleInteger )
6 import Data.Char ( isSpace )
7
8 data Number = NumZ Integer | NumQ (Ratio Integer) | NumR Double | NumFD { x ::
    Double, x' :: Double }
9
10 limit_denominator :: Number → Number → Number
11 limit_denominator (NumZ lim) (NumQ frac@(n:%d)) =
12     if lim <= 0 then NumZ 0 else
13         if d <= lim then NumQ frac else
14             NumQ $ limit (0, 1, 1, 0) lim frac
15 where
16     limit (p0, q0, p1, q1) lim frac@(n:%d) =
17         let a = n `quot` d
18             q2 = q0 + a * q1
19         in if q2 > lim
20             then
21                 let k = (lim - q0) `quot` q1
22                     bound1 = (p0 + k * p1) % (q0 + k * q1)
23                     bound2 = p1 % q1
24                 in if abs (bound2 - frac) <= abs (bound1 - frac)
25                     then bound2 else bound1
26             else
27                 limit (p1, q1, p0 + a * p1, q2) lim (d :% (n - a * d))

```

C

```

int nesimo_primo(int n) {
    int contador_primos = 0; // Cantidad de primos encontrados hasta el momento
    int candidato = 2; // Numero que estamos probando
    while (contador_primos < n) {
        if (es_primo(candidato)) { // Si el numero es primo
            contador_primos++; // Encontre un primo mas
        }
        candidato++; // Pruebo con el siguiente numero
    }
    return candidato - 1; // El ultimo numero que probe es el n-esimo primo
}

```

Python

```
1 def nesimo_primo(n):  
2     contador_primos = 0 # Cantidad de primos encontrados hasta el momento  
3     candidato = 2 # Numero que estamos probando  
4     while contador_primos < n:  
5         if es_primo(candidato): # Si el numero es primo  
6             contador_primos += 1 # Encontre un primo mas  
7             candidato += 1 # Pruebo con el siguiente numero  
8     return candidato - 1 # El ultimo numero que probe es el n-esimo primo
```

Frames

Teorema 1.1

Sea f una función continua en el intervalo $[a, b]$. Entonces, f es integrable en $[a, b]$.

Ejercicio 1.2

Sea f una función continua en el intervalo $[a, b]$. Entonces, f es integrable en $[a, b]$.

Observación S

ea f una función continua en el intervalo $[a, b]$. Entonces, f es integrable en $[a, b]$.