```python
#Generador Tausworthe

#Los iteradores en Python son un objeto que puede iterar como tipos de
datos de
# secuencia como list, tuple, str, etc.
from itertools import zip_longest

r = int(input("valor de r: "))
q = int(input("valor de q: "))
binarios = int(input("Digite el valor de la base: "))

def operacionXOR(a,b):
  ab = 1
  if a  == b:
    ab = 0
  return ab

bits = []
b = (2**q) - 1

for i in range(0,b):
  bits.append(0)

print(bits)

for i in range(0,q):
  bits[i] = 1
  bits.append(1)

print(bits)

a = q+1
for i in range(a,len(bits)):
  i1 = i - r
  i2 = i - q
  bits[i] = operacionXOR(bits[i1],bits[i2])

print(bits)

def binarioADecimal(binario):
  a = 0
  p = (binarios-1)
  for i in range(0,len(binario)):
    if binario[i] == 1:
      a += 2**(p-i)
  return a
test_list = bits

def elementos(n, iterable, padvalue='1'):
    return zip_longest(*[iter(iterable)]*n, fillvalue=padvalue)

print("\n","It.", "\t", "Base 2", "\t","Base 10","\t","Ui","\n")
d = 0
for output in elementos(binarios,test_list):
    lst_new = [str(a) for a in output]
    print(d,"\t" ," ".join(lst_new), "\t", binarioADecimal(output), "\
t", "\t", binarioADecimal(output)/(2**binarios))
    d +=1
```

```
valor de r: 1
valor de q: 3
Digite el valor de la base: 2
[0, 0, 0, 0, 0, 0, 0]
[1, 1, 1, 0, 0, 0, 0, 1, 1, 1]
[1, 1, 1, 0, 1, 0, 0, 1, 1, 1]

 It.   Base 2     Base 10    Ui

0      1 1   3          0.75
1      1 0   2          0.5
2      1 0   2          0.5
3      0 1   1          0.25
4      1 1   3          0.75
```