

Considere o programa, Python 3, que se segue. Escreva a classe `Reverser`. Os objetos da classe `Reverser` tem o método `reverse()`. O método `reverse()` reverte todos os números que não possuam mais de 32 bits, ou seja, os números positivo com mais de 32 bits irão reverter para números negativos e vice-versa.

Escreva ainda as funções `to_bits()` e `binary_sum()`, estas não são métodos da classe acima. A função `to_bits()` converte um número decimal em binário e a função `binary_sum()` soma dois números binários.

```
from random import randint
from random import seed

seed(135)

class Reverser:

    def reverse(self, num):

def to_bits(num):

def binary_sum(num1, num2):

a = []
b = []
c = []

dim = 13
for i in range(dim):
    a.append(randint(-2**31, 2**31))

r = Reverser()
for i in range(dim):
    b.append(r.reverse(a[i]))

for i in range(dim):
    c.append(r.reverse(b[i]))
```

Considere o código de testes, à classe `Reverser` e funções `to_bits()` e `binary_sum()`, que se segue.

```
>>> r.reverse("-110101101101001011111110011000")
"-000110011111110100101101101011"
>>> r.reverse("101011011001101010010111101100")
"001101111010010101100110110101"
>>> r.reverse(728147436)
634741827
>>> to_bits(0)
"0"
>>> to_bits(728147436)
"101011011001101010010111101100"
>>> binary_sum("101011011001101010010111101100"
               , "10101001100001010111001111110")
"10000000100101110101010001101010"
>>> binary_sum("101011011001101010010111101100"
               , "-1100001000110110110010110000100")
"-110101101101001011111110011000"
```

Acrescente a este programa o código que lhe permita indicar se as afirmações seguintes são verdadeiras ou falsas.

Indique se é verdadeiro ou falso.