THE GOOD, THE BAD AND THE UGLY (2138)

https://acm.timus.ru/problem.aspx?space=1&num=2138

PHUNTSHO WANGDI

DIFFICULTY LEVEL : 235

Everyone knows that computer see integers not in the same way as humans do. Instead of digits from 0 to 9 they use digits from 0 to 255. For example, the integer 1 000 000 can by represented by a computer with three digits 15, 66, 64 (let's denote it as 15;66;64), because $15 \cdot 256^2 + 66 \cdot 256 + 64 = 1\,000\,000$. On top of that, integers in computers have a fixed size. In this problem the size of an integer is always 4 digits, so the integer 1 000 000 will be represented as 0;15;66;64. Computers use this exact format to communicate with each other.

This system may seem strange, but it works. Or, it used to work, until an evil genius reversed the order of digits in half of the computers! These computers now interpret 0;15;66;64 not as 1 000 000 but as 1 078 071 040 ($64 \cdot 256^3 + 66 \cdot 256^2 + 15 \cdot 256 + 0 = 1\,078\,071\,040$). No one knows how to fix it, so the computers that are left untouched are now called "good", and other computers are called "bad". In order for "good" and "bad" computers to communicate, the integers that are read incorrectly must be translated into correct integers.

For example, let A and B be two computers of opposite types. Let's say the computer A wants to send the computer B a non-negative integer not exceeding 4 294 967 295. The computer A writes this integer down with four digits from 0 to 255 and sends them to the computer B. Computer B reads the received digits as v, which doesn't necessary match the integer the computer A was trying to send.

Write a program that would help a "good" or a "bad" computer B determine, what integer the computer A of the opposite type tried to send.

https://acm.timus.ru/problem.aspx?space=1&num=2138

METHODOLOGY

Number system conversion.

Hex to Binary

Binary to Decimal

Decimal to Binary

STEPS FOR CONVERTING DECIMAL TO HEXADECIMAL

Step 1: If the given decimal number is less than 16, the hex equivalent is the same. Remembering that the letters A, B, C, D, E and F are used for the values 10, 11, 12, 13, 14 and 15, convert accordingly. For example, the decimal number 15 will be F in hex.

Step 2: If the given decimal number is 16 or greater, divide the number by 16.

Step 3: Write down the remainder.

Step 4: Divide the part before the decimal point of your quotient by 16 again. Write down the remainder.

STEPS FOR CONVERTING DECIMAL TO HEXADECIMAL

Step 5: Continue this process of dividing by 16 and noting the remainders until the last decimal digit you are left with is less than 16.

Step 6: When the last decimal digit is less than 16, the quotient will be less than 0 and the remainder will be the digit itself.

Step 7: The last remainder you get will be the most significant digit of your hex value while the first remainder from Step 3 is the least significant digit. Therefore, when you write the remainders in reverse order - starting at the bottom with the most significant digit and going to the top- you will reach the hex value of the given decimal number.

EXAMPLE
CONVERTING DECIMAL TO HEX

501_{10} convert to hexadecimal.

501/16 = 31.3125, Remainder = 0.3125 *16 = 5

31/16 = 1.9375 , Remainder = 0.9375*16 = 15

1/16 = 0.0625, Remainder = 0.0625*16 = 1

$510_{10} = 1F5_{16}$

CONDITIONS

Instead of 16 which is for hexadecimal we have 256.

We use same method of conversion that we used for converting from decimal to hexadecimal.

```
from math import modf, pow

comp = input()
x = int(input())

def interprete():
    global x
    r = []
    if x>=256:
        while x>=256:
            x=x/256
            b,a = modf(x)
            x=a
            remainder = b*256
            r.append(remainder)
        r.append(a)

    else:
        r.append(x)
    return r

out = 0
r=interprete()

if len(r)<4:
    for i in range(4-len(r)):
        r.append(0)
for i in range(len(r)):
    out = out + r[i]*pow(256,len(r)-(i+1))

print(int(out))
```

8656963	01:05:02 28 Nov 2019	Phuntsho Wangdi	2138. The Good, the Bad and the Ugly	Python 3.6	Accepted

THANK YOU