1295. Crazy Notions

Tanapat L. 6014216
Aung Hlaing Moe 5745402
How much zeros does its decimal notation end with?

100  answer is 2

640  answer is 1

999  answer is 0
**Input**

The only line contains an integer $n$ ($1 \leq n \leq 300000$).

**Output**

Output the number of zeroes the decimal notation of $1^n + 2^n + 3^n + 4^n$ ends with.

<table>
<thead>
<tr>
<th>input</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
n = input()
x = 1**int(n) + 2**int(n) + 3**int(n) + 4**int(n)
arr = list(str(x))
count = 0
for i in range(len(arr)-1,-1,-1):
    if arr[i] == '0':
        count += 1
    else:
        break
print(count)
Running Time of My first attempt

\[ 2^{**n} = 2 \times 2 \times 2 \times \ldots \times 2 \]

\[ 2^{**n} + 3^{**n} + 4^{**n} = 3n \]
```python
from time import time

n = 1000000

x = 2**n
y = 3**n

start = time()
print(x+y)

end = time()
print("time:", end-start)
```
\[2^{n} + 3^{n}\]

\[n = 10000000\]

\[n = 100000\]
Thank you for hint

if \{(1^n + 2^n + 3^n + 4^n) \mod 10\} is equal to 0, then we find 1 zero.
if \{(1^n + 2^n + 3^n + 4^n) \mod 100\} is equal to 0, then we find 2 zeros.
and so on....

now, how to calculate \{(1^n + 2^n + 3^n + 4^n) \mod 10\}:

Look,
\{(1^n + 2^n + 3^n + 4^n) \mod 10\}
= \{(1^n \mod 10) + (2^n \mod 10) + (3^n \mod 10) + (4^n \mod 10)\} \mod 10 \quad [\text{simple modulo equivalencies}]

now,
(4^n \mod 10)
= (((((4\%10)*4\%10)*4\%10)*4\%10)\%10 ... ... ... (n times) \quad [4\%10, \text{then multiply by 4, then mod 10, loop for n times}]
similarly for (2^n \mod 10) and (3^n \mod 10). No need for 1, because (1^n \mod 10) is always 1.

In this way, calculate the result of \{(1^n + 2^n + 3^n + 4^n) \mod m\} for m = 10, 100, ans so on... and count zeros... :)}
n = input()

cal1 = cal2 = cal3 = cal4 = 1

for i in range(int(n)):
    cal2 = (cal2*2)%100
    cal3 = (cal3*3)%100
    cal4 = (cal4*4)%100

x = cal1+cal2+cal3+cal4

arr = list(str(x))

count = 0
for i in range(len(arr)-1,-1,-1):
    if arr[i] == '0':
        count += 1
    else:
        break

print(count)
\[2^{**2} = 4\]
\[2^{**22} = ...04\]
\[3^{**1} = 3\]
\[3^{**21} = ...03\]
n = 0 2
n = 0 3
n = 0 4
n = 1 4
n = 1 9
n = 1 16
n = 21 4
n = 21 9
n = 21 16