Divisiblity by 3

Divisiblity by 3 rule is pretty simple rule: Given a number sum the digits of number and check if sum is divisible by 3.If divisible then it is divisible by 3 else not divisible.Seems pretty simple but what if we want to extend this rule in binary representation!!

Given a binary representation we can again find if it is divisible by 3 or not. Making it little bit interesting what if only length of binary representation of a number is given say n.

Now can we find how many numbers exist in decimal form(base 10) such that when converted into binary(base 2) form has n length and is divisible by 3 ?? (1 <= n < 2*10^18)

Input

Length of binary form: n

output

Print in new line the answer modulo 1000000007.

Example

Input:
1

2

Output:
1

2

Explanation: For n=2 there are only 2 numbers divisible by 3 viz.0 (00) and 3 (11) and having length 2 in binary form.

NOTE: There are multiple testfiles containing many testcases each so read till EOF.

Warnings: Leading zeros are allowed in binary representation and slower languages might require fast i/o. Take care.