#include <iostream>

#include<climits>

using namespace std;

//TOP-DOWN

//Minimum steps to one

//n->n/3, n->n/2, n=n-1

int minSteps(int n,int dp[]){

//Base case

if(n==1){

return 0;

}

//Recursive case

//Lookup if n is already computed?

if(dp[n]!=0){

return dp[n];

}

//Compute if dp[n] is not known (for the first time)

int op1,op2,op3;

op1=op2=op3=INT\_MAX;

if(n%3==0){

op1=minSteps(n/3,dp);

}

if(n%2==0){

op2=minSteps(n/2,dp);

}

op3=minSteps(n-1,dp);

int ans=min(op1,min(op2,op3))+1;

return dp[n]=ans;

}

//BOTTOM-UP

int minBUSteps(int n){

int dp[100]={0};

dp[1]=0;

for(int i=2;i<=n;i++){

int op1,op2,op3;

op1=op2=op3=INT\_MAX;

if(n%3==0){

op1=dp[i/3];

}

if(n%2==0){

op2=dp[i/2];

}

op3=dp[i-1];

dp[i]=min(op1,min(op2,op3))+1;

}

return dp[n];

}

int main() {

int n;

cin>>n;

int dp[100]={0};

cout<<minSteps(n,dp)<<endl;

cout<<minBUSteps(n)<<endl;

return 0;

}

Input-

10

Output-

3

3