class Solution {

public:

int divide(int dividend, int divisor) {

int dividend\_sign = 1;

int divisor\_sign = 1;

long long num\_dividend = dividend;

long long num\_divisor = divisor;

long long ret = 0;

if (divisor == 0)

return 0;

if (dividend == 0)

return 0;

if (num\_dividend < 0)

{

dividend\_sign = -1;

num\_dividend = -num\_dividend;

}

if (num\_divisor < 0)

{

divisor\_sign = -1;

num\_divisor = -num\_divisor;

}

if (num\_divisor == 1)

ret = num\_dividend;

else

{

while (num\_dividend >= num\_divisor)

{

long long temp = num\_divisor;

int i = 0;

while (num\_dividend >= temp)

{

ret += (1 << i);

num\_dividend -= temp;

temp <<= 1;

i++;

}

}

}

if (ret > INT\_MAX && (dividend\_sign \* divisor\_sign > 0))

return INT\_MAX;

if (dividend\_sign \* divisor\_sign < 0)

return -1 \* ret;

return ret;

}

};