[**Divide Two Integers**](https://leetcode.com/problems/divide-two-integers/)

**public** **class** DivideIntegers {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

System.***out***.println(*divide*(7 , -3));

}

**public** **static** **int** divide(**int** dividend, **int** divisor) {

**if**(divisor == 0) {

**return** 0;

}

**int** sign = 1;

**if**((dividend ^ divisor) < 0) {

sign = -sign;

}

**long** ldivisor = Math.*abs*((**long**)divisor);

**long** ldividend = Math.*abs*((**long**)dividend);

**if**(ldivisor == 0 || ldividend < ldivisor) {

**return** 0;

}

**long** lresult = *findDivision*(ldividend, ldivisor);

**if**(lresult > Integer.***MAX\_VALUE***) {

**return** sign == 1 ? Integer.***MAX\_VALUE*** : Integer.***MIN\_VALUE***;

}

**return** sign == 1 ? (**int**)lresult : (**int**)(-lresult);

}

**public** **static** **long** findDivision(Long ldividend, Long ldivisor) {

**if**(ldividend < ldivisor) {

**return** 0;

}

**long** sum = ldivisor;

**long** multiple = 1;

**while**(ldividend >= (sum + sum)) {

sum \*= 2;

multiple \*= 2;

}

**return** multiple + *findDivision*(ldividend - sum, ldivisor);

}

}

Time Complexity : O(logn) , n is dividend

Space Complexity : O(1), constant space