Pow(x, n)

**class** **Solution** {

**public** **double** **myPow**(**double** x, **int** n) {

**long** n\_long = n;

**return** helper(x, n\_long);

}

**private** **double** **helper**(**double** x, **long** n\_long) {

**if**(n\_long == 0) {

**return** 1;

}

**if**(n\_long < 0) {

n\_long = -n\_long;

x = 1/x;

}

**return** (n\_long%2 == 0) ? helper(x\*x, n\_long/2) : x \* helper(x\*x, n\_long/2);

}

}

**class** **Solution** {

**public** **double** **myPow**(**double** x, **int** n) {

**long** nL = (**long**) n;

**return** pow(x, nL);

}

**private** **double** **pow**(**double** x, **long** n) {

**if** (n == 0) **return** 1;

**if** (n < 0) **return** 1.0 / pow (x, -n);

**double** res = 1;

**while** (n != 0) {

**double** curRes = x;

**long** cnt = 1;

**while** (n > 2 \* cnt) {

curRes \*= curRes;

cnt += cnt;}

res \*= curRes;

n = n - cnt;}

**return** res;}}