**[Integer to English Words](https://leetcode.com/problems/integer-to-english-words/)**

**package** leetcodeHard;

**public** **class** IntegerToWords {

**public** **static** String[] *LESS\_THAN\_TENS* = {"" , "One", "Two", "Three" , "Four", "Five" , "Six", "Seven",

"Eight", "Nine"};

**public** **static** String[] *LESS\_THAN\_TWENTY* = {"Ten", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen",

"Sixteen", "Seventeen", "Eighteen", "Nineteen"};

**public** **static** String[] *LESS\_THAN\_HUNDRED* = {"", "Ten","Twenty","Thirty","Forty","Fifty","Sixty",

"Seventy","Eighty","Ninety"};

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

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

System.***out***.println(*numberToWords*(123456));

}

**public** **static** String numberToWords(**int** num) {

**if**(num == 0) {

**return** "Zero";

}

**return** *helper*(num);

}

**public** **static** String helper(**int** num) {

String result = "";

**if**(num < 10) {

result += *LESS\_THAN\_TENS*[num];

}

**else** **if**(num < 20) {

result += *LESS\_THAN\_TWENTY*[num%10];

}

**else** **if**(num < 100) {

result += *LESS\_THAN\_HUNDRED*[num/10] +" " + *LESS\_THAN\_TENS*[num%10];

}

**else** **if**(num < 1000) {

result += *helper*(num/100) +" Hundred " + *helper*(num%100);

}

**else** **if**(num < 1000000) {

result += *helper*(num/1000) +" Thousand " + *helper*(num%1000);

}

**else** **if**(num < 1000000000) {

result += *helper*(num/1000000) +" Million " + *helper*(num%1000000);

}

**else** {

result += *helper*(num/1000000000) +" Billion " + *helper*(num%1000000000);

}

**return** result.trim();

}

}

Time Complexity : O(1) constant time

Space Complexity : O(1) constant space