public class NumberToWord

{

private static final String[] specialNames = {

"",

" thousand",

" million",

" billion",

" trillion",

" quadrillion",

" quintillion"

};

private static final String[] tensNames = {

"",

" ten",

" twenty",

" thirty",

" forty",

" fifty",

" sixty",

" seventy",

" eighty",

" ninety"

};

private static final String[] numNames = {

"",

" one",

" two",

" three",

" four",

" five",

" six",

" seven",

" eight",

" nine",

" ten",

" eleven",

" twelve",

" thirteen",

" fourteen",

" fifteen",

" sixteen",

" seventeen",

" eighteen",

" nineteen"

};

private String convertLessThanOneThousand(int number) {

String current;

if (number % 100 < 20){

current = numNames[number % 100];

number /= 100;

}

else {

current = numNames[number % 10];

number /= 10;

current = tensNames[number % 10] + current;

number /= 10;

}

if (number == 0) return current;

return numNames[number] + " hundred" + current;

}

public String convert(int number) {

if (number == 0) { return "zero"; }

String prefix = "";

if (number < 0) {

number = -number;

prefix = "negative";

}

String current = "";

int place = 0;

do {

int n = number % 1000;

if (n != 0){

String s = convertLessThanOneThousand(n);

current = s + specialNames[place] + current;

}

place++;

number /= 1000;

} while (number > 0);

return (prefix + current).trim();

}

public static void main(String[] args) {

NumberToWord obj = new NumberToWord();

System.out.println("\*\*\* " + obj.convert(123456789));

System.out.println("\*\*\* " + obj.convert(-55));

}

}