Interger to English

Convert a non-negative integer `num` to its English words representation.

Example 1:

```
Input: num = 123
Output: "One Hundred Twenty Three"
```

Example 2:

```
Input: num = 12345
Output: "Twelve Thousand Three Hundred Forty Five"
```

Example 3:

```
Input: num = 1234567
Output: "One Million Two Hundred Thirty Four Thousand Five Hundred Sixty Seven"
```

Constraints:

- `0 <= num <= 2^31 - 1`

My code

```
#!/usr/bin/env python3
import sys

def makeHundreads(x):
    x.value = x.value + " Hundred"
    return (x)

class Solution:
    def numberLessThousandToWords(self, num: int) -> str:
        usual_numbers = {
            1:"One",
            2:"Two",
            3:"Three",
            4:"Four",
            5:"Five",
            6:"Six",
```

```
            7:"Seven",
            8:"Eight",
            9:"Nine",
            10:"Ten",
            11:"Eleven",
            12:"Twelve",
            13:"Thirteen",
            14:"Fourteen",
            15:"Fifteen",
            16:"Sixteen",
            17:"Seventeen",
            18:"Eighteen",
            19:"Nineteen",
            20:"Twenty",
            30:"Thirty",
            40:"Forty",
            50:"Fifty",
            60:"Sixty",
            70:"Seventy",
            80:"Eighty",
            90:"Ninety"
            # si je met one hundred je devrais mettre les autres... essayons sans pour voir deja
        }
        unit_map = {
            1:"One",
            2:"Two",
            3:"Three",
            4:"Four",
            5:"Five",
            6:"Six",
            7:"Seven",
            8:"Eight",
            9:"Nine"
        }

        ten_map = {
            1:"Ten",
            2:"Twenty",
            3:"Thirty",
            4:"Forty",
            5:"Fifty",
            6:"Sixty",
            7:"Seventy",
            8:"Eighty",
            9:"Ninety"
        }

#        hundread_map = dict(map(makeHundreads, unit_map.))
        hundread_map = {
            1:"One Hundred",
            2:"Two Hundred",
            3:"Three Hundred",
            4:"Four Hundred",
            5:"Five Hundred",
            6:"Six Hundred",
            7:"Seven Hundred",
```

```
            8:"Eight Hundred",
            9:"Nine Hundred"
        }
        numbers_map = {
            100: hundread_map,
            10 : ten_map,
            1 : unit_map
        }
        if (num in usual_numbers.keys()):
            return (usual_numbers[num])
        string = ""
        decimal = 1
        while (num / (decimal * 10) >= 1):
            decimal *= 10
        while (decimal >= 1):
            quotient = num // decimal
            if (num in usual_numbers.keys()):
                string += usual_numbers[num]
                return (string)
            if (quotient in numbers_map[decimal]):
                string += numbers_map[decimal][quotient]
                if decimal / 10 >= 1:
                    string += " "
            num %= decimal
            decimal /= 10
        return (string)
    def numberToWords(self, num: int) -> str:
        usual_numbers = {
            0:"Zero",
            1:"One",
            2:"Two",
            3:"Three",
            4:"Four",
            5:"Five",
            6:"Six",
            7:"Seven",
            8:"Eight",
            9:"Nine",
            10:"Ten",
            11:"Eleven",
            12:"Twelve",
            13:"Thirteen",
            14:"Fourteen",
            15:"Fifteen",
            16:"Sixteen",
            17:"Seventeen",
            18:"Eighteen",
            19:"Nineteen",
            20:"Twenty",
            30:"Thirty",
            40:"Forty",
            50:"Fifty",
            60:"Sixty",
            70:"Seventy",
            80:"Eighty",
            90:"Ninety"
```

```
            # si je met one hundred je devrais mettre les autres... essayons sans pour voir deja
        }
        hundread = "Hundred"
        thousand_array = ["", " Thousand", " Million", " Billion"]
        if (num in usual_numbers.keys()):
            return (usual_numbers[num])
        power = 0
        power_int = 1
        string = ""
        while (num / (power_int * 1000) >= 1):
            power_int *= 1000
            power += 1
        while (power >= 0):
            quotient = num // power_int
            #print(self.numberLessThousandToWords(quotient))
            ret = self.numberLessThousandToWords(quotient)
            string += ret
            if (ret != ""):
                string += thousand_array[power]
            power -= 1
            num %= power_int
            power_int /= 1000
            string += " "
        string = " ".join(filter(None, string.split(" ")))
        return (string.strip())

if (len(sys.argv) == 1):
    exit(84)
sol = Solution()
print(sol.numberToWords(int(sys.argv[1])))
```