

## 273. Integer to English Words

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:

- $0 \leq \text{num} \leq 2^{31} - 1$

## 273. Integer to English Words

```
const vector<string> to_19{
    "", "One", "Two", "Three", "Four",
    "Five", "Six", "Seven", "Eight", "Nine",
    "Ten", "Eleven", "Twelve", "Thirteen", "Fourteen",
    "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen"};

const vector<string> tens{
    "", "", "Twenty", "Thirty", "Forty",
    "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"};
```

```

/*
    m: number of digits of num
    Time complexity: O(logm)
    Space complexity: O(logm)
*/
class Solution {
public:
    const vector<string> to_19{
        "", "One", "Two", "Three", "Four",
        "Five", "Six", "Seven", "Eight", "Nine",
        "Ten", "Eleven", "Twelve", "Thirteen", "Fourteen",
        "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen"};

    const vector<string> tens{
        "", "", "Twenty", "Thirty", "Forty",
        "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"};

```

```

public:
    std::string numberToWords(int num) {
        if(num==0) return "Zero";

```

```

        auto solve=[&](int n,auto& self)->std::string{
            std::string s;
            if(n<20) s=to_19[n];
            else if(n<100) s=tens[n/10]+" "+to_19[n%10];
            else if(n<1000) s=self(n/100,self)+" Hundred "+self(n%100,self);
            else if(n<1000000) s=self(n/1000,self)+" Thousand "+self(n%1000,self);
            else if(n<1000000000) s=self(n/1000000,self)+" Million "+self(n%1000000,self);
            else s=self(n/1000000000,self)+" Billion "+self(n%1000000000,self);

            s.erase(0, s.find_first_not_of(' '));
            s.erase(s.find_last_not_of(' ') + 1);

            return s;
        };

```

```

        return solve(num,solve);

```

```

    }

```

```

};

```