

PAT_A1005.Spell It Right(20)

PAT_A1005.Spell It Right(20)

- 1. Abstraction
 - 1.1 Method and idea
 - 1.2 Notice
- 2. Problem
- 3. Code
 - 3.1 Edit 0
 - 3.1.1 Algorithm abstraction
 - 3.1.2 Notice
 - 3.2 Edit 1
 - 3.2.1 Algorithm abstraction
 - 3.2.2 Notice

1. Abstraction

1.1 Method and idea

- None

1.2 Notice

- None

2. Problem

Given a non-negative integer N, your task is to compute the sum of all the digits of N, and output every digit of the sum in English.

Input Specification:

Each input file contains one test case. Each case occupies one line which contains an N (≤ 10100).

Output Specification:

For each test case, output in one line the digits of the sum in English words. There must be one space between two consecutive words, but no extra space at the end of a line.

Sample Input:

```
12345
```

Sample Output:

```
one five
```

3. Code

3.1 Edit 0

3.1.1 Algorithm abstraction

字符串读入，遍历求和，转换为字符串，遍历输出

3.1.2 Notice

- None

```
#include <iostream>
#include <string>
#include <algorithm>
using std::cin;
using std::cout;
using std::endl;
using std::string;
using std::fill;

int how_many_bit(int sum){
    int ans=0;
    while(sum){
        sum=sum/10;
        ++ans;
    }
    return ans;
}

int ans[11]={0};

int get_every_bit(int sum,int bit_num){
    int temp=sum;
    for(int i=0;i<bit_num;++i){
        int yu=temp%10;
        ans[i]=yu;
        temp=temp/10;
    }
}

int main(){
    string num;
    cin>>num;
    const int maxn=10;
    int bit[maxn];
    fill(bit,bit+maxn,0);
    int sum=0;
    for(int i=0;i<(int)num.size();++i){
        ++bit[num[i]-'0'];
    }

    for(int i=0;i<10;++i){
```

```

        sum+=bit[i]*i;
    }
    if(sum==0){
        cout<<"zero"<<endl;
        return 0;
    }
    int bit_num=how_many_bit(sum);

    get_every_bit(sum,bit_num);
    for(int i=bit_num-1;i>=0;--i){
        int bit=ans[i];
        string s;
        switch(bit){
            case 0:
                s=string("zero");
                break;
            case 1:
                s=string("one");
                break;
            case 2:
                s=string("two");
                break;
            case 3:
                s=string("three");
                break;
            case 4:
                s=string("four");
                break;
            case 5:
                s=string("five");
                break;
            case 6:
                s=string("six");
                break;
            case 7:
                s=string("seven");
                break;
            case 8:
                s=string("eight");
                break;
            case 9:
                s=string("nine");
                break;
        }
        cout<<s;
        if(i!=0){
            cout<<" ";
        }
        else{
            cout<<endl;
        }
    }

    return 0;

```

```
}
```

3.2 Edit 1

3.2.1 Algorithm abstraction

字符串读入，遍历求和，转换为字符串，遍历输出

3.2.2 Notice

- None

```
#include <iostream>
#include <string>
#include <sstream>

using std::cin;
using std::cout;
using std::endl;
using std::string;
using std::stringstream;
using std::ios;
void print_result(string result){
    for(int i=0;i<(int)result.size();++i){
        switch(result[i]){
            case '0':
                cout<<"zero";
                break;
            case '1':
                cout<<"one";
                break;
            case '2':
                cout<<"two";
                break;
            case '3':
                cout<<"three";
                break;
            case '4':
                cout<<"four";
                break;
            case '5':
                cout<<"five";
                break;
            case '6':
                cout<<"six";
                break;
            case '7':
                cout<<"seven";
                break;
```

```

        case '8':
            cout<<"eight";
            break;
        case '9':
            cout<<"nine";
            break;
    }
    if(i!=(int)result.size()-1){
        cout<<" ";
    }
    else{
        cout<<endl;
    }
}
}
int main(){
    ios::sync_with_stdio(false);
    string num_s;
    cin>>num_s;
    int sum=0;
    for(int i=0;i<(int)num_s.size();++i){
        sum+=(num_s[i]-'0');
    }
    stringstream convert;
    convert<<sum;
    string result;
    convert>>result;
    print_result(result);
    return 0;
}

```