

运用向量求两个很长正数的和

题目要求：

实现一个加法器，使其能够输出a+b的值。

输入：

输入包括两个数a和b，其中a和b的位数不超过1000位。

输出：

可能有多组测试数据，对于每组数据，
输出a+b的值。

样例输入：

```
2 6  
10000000000000000000 10000000000000000000000000000000
```

样例输出：

```
8  
10000000000100000000000000000000
```

我的实现：

```
#include <iostream>
#include <vector>
using namespace std;

int min(int a, int b){
    return a > b ? a : b;
}

int main(){
    vector<int>v;

    string str1 = "";
    string str2 = "";
    int temp = 0; //用来储存进位
    int len1, len2;
    int i = 0, j = 0, k = 0;
    int sum = 0;
    while(cin >> str1 >> str2){
        v.clear();
        len1 = str1.length();
        len2 = str2.length();
        i = len1 - 1;
        j = len2 - 1;

        len1 = max(len1, len2);
```

```
temp = 0;
while(i >= 0 && j >= 0){
    sum = str1[i] - '0' + str2[j] - '0' + temp;
    if(sum > 9){
        temp = sum / 10; //进位
    }
    //cout << sum << endl;
    v.insert(v.begin(), sum % 10);
    i--;
    j--;
}
```

```
if(i < 0){
    while(j >= 0){
        sum = str2[j] - '0' + temp;
        temp = 0;
        if(sum > 9){
            temp = sum / 10; //进位
        }
        v.insert(v.begin(), sum % 10);
        j--;
    }
}
```

```
if(j < 0){
    while(i >= 0){
        sum = str1[i] - '0' + temp;
        temp = 0;
        if(sum > 9){
            temp = sum / 10; //进位
        }
        v.insert(v.begin(), sum % 10);
        i--;
    }
}
```

```
if(temp > 0){
    v.insert(v.begin(), temp);
}
```

```
for(i = 0; i < v.size(); i++)
    cout << v[i];
cout << endl;
}
```

}

```
9 999999999999999999
10000000000000000000
99999999 9
1000000000
99 99
198
10000000000000000000 110
10000000000000000000110
```

运行结果：

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接：[【】（）](#)