

## 高精度加法

示例：计算 $12345+999999$

## 算法思路

1、将数字按字符串进行逆序存储，我们可以得到两个char数组

数组A	5	4	3	2	1	
数组B	9	9	9	9	9	9

2、取二者中长度最大者作为计算终止长度，很明显，这里的终止长度应该为6.

3、将字符串数组转为数字数组，方法就是使用数组中的每一个字符减去'0'，进而我们得到了两个新的数组

数组A	5	4	3	2	1	
数组B	9	9	9	9	9	9

4、从索引0开始对位相加，同时需要处理进位

数组A	5	4	3	2	1				
数组B	9	9	9	9	9	9			
结果	14	13	12	11	10	9			

5	4	3	2	1		
9	9	9	9	9	9	
4	4	3	2	1	0	1

5、如果最高位不等于0，则需要把长度进行增加，很明显，当前的长度应该为7

6、可能会存在加数本身待有前导0的情况，因此我们需要使用while去除前导0

7、因为我们存储数组的时候是按照逆序进行存储的，因此，在打印的时候，我们需要将其修正过来，因此我们同样需要逆序打印。

结果	1	0	1	2	3	4	4
----	---	---	---	---	---	---	---

## 编码

```
#include<bits/stdc++.h>

using namespace std;

char s1[201], s2[201]; //string也是可以的
int a[201], b[201]; //实际参与计算的数组
int len;

void HighAccuracyAlgorithm() {
    int l1 = strlen(s1); // "strlen"是一个计算字符串长度的函数
    int l2 = strlen(s2); // 将输入的两个字符串的长度赋值给l1, l2
```

```

    len = max(l1, l2); //求出最大长度
//两个for循环是将输入的两个字符串逆序存储
for (int i = l1 - 1; i >= 0; i--)
    a[l1 - i - 1] = s1[i] - '0'; //将索引进行逆序
for (int i = l2 - 1; i >= 0; i--)
    b[l2 - i - 1] = s2[i] - '0';
//因为大数与大数是一位一位运算的，还要涉及进位等
for (int i = 0; i < len; i++) {
    a[i] = a[i] + b[i]; //运算
    a[i + 1] += a[i] / 10; //向高位进位
    a[i] = a[i] % 10; //保留实际数字
}
//一定要判断最高位是否进位
if (a[len] != 0) {
    len++;
}
//去除前导0
while (a[len - 1] == 0 && len > 1) {
    len--;
}
//逆序打印
string res = "";
for (int i = len - 1; i >= 0; i--) //输出结果
{
    res += a[i] + '0';
}
//打印字符串
printf("%s", res.c_str());
}

int main() {
    cin >> s1 >> s2;
    HighAccuracyAlgorithm();
    return 0;
}

```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

