



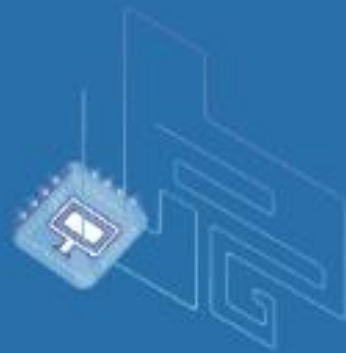
学习目标



1. 学会高精度加法的基本使用
2. 学会高精度减法的基本使用



知识讲授





高精度加减法

高精度的加减法，主要是用于求解数据特别大的相关题目的结果。比如让你求两个数字的和，但是两个数字数位都超过了1000位。这种情况下我们就没有办法用int或者long long去存储数据了，因此我们可以用字符串或者字符数组来存放两个数字来解决。

高精度的加减法，和我们数学上的列竖式的原理是一致的，接下来我们就一起来看一看其中的奥秘吧。



高精度加法实例

大数字加法，两个非负数字相加。

 参考代码

```
#include <iostream>
#include <cmath>
using namespace std;
string s1,s2;
int a[10001],b[10001],c[10001];
void Plus();
int main(){
    cin>>s1>>s2;
    a[0]=s1.size();  b[0]=s2.size();
    for(int i=1;i<=a[0];i++)    a[i]=s1[a[0]-i]-'0';
    for(int i=1;i<=b[0];i++)    b[i]=s2[b[0]-i]-'0';
    Plus();
    for(int i=c[0];i>=1;i--)  cout<<c[i];
    cout<<endl;
    return 0;
}
```

参考代码

```
void Plus()
{
    c[0]=max(a[0],b[0]);
    int jw=0,k;
    for(int i=1;i<=c[0];i++) { //相加
        k=a[i]+b[i]+jw; //相加结果
        c[i]=k%10; //存进结果数组
        jw=k/10; //计算进位
    }
    if(jw!=0) { //处理最高位的进位
        c[0]++;
        c[c[0]]=jw;
    }
}
```

高精度减法实例

两个非负大数相减。结果为负数就输出负数。

参考代码

```
#include <iostream>
#include <cmath>
using namespace std;
string s1,s2,p1,p2;
int a[100001],b[100001],c[100001],flag;
void Sub()
{
    c[0]=a[0]; //保证a是比较大的数字
    for(int i=1;i<=c[0];i++){ //相减
        if(a[i]<b[i]){
            a[i]+=10;
            a[i+1]--;
        }
        c[i]=a[i]-b[i];
    }
    while(c[c[0]]==0&& c[0]>1) //处理开头的0,并防止结果为0
        c[0]--;
}
```

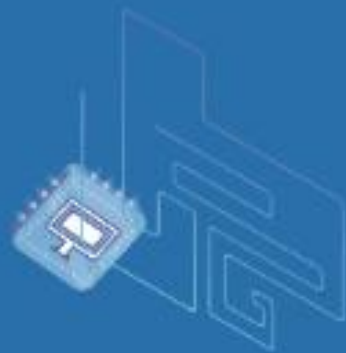
参考代码

```
int main()
{
    cin>>s1>>s2;
    if(s1.size()<s2.size()){ //保证s1比较大
        flag=1; //标记,结果为负输出负号
        swap(s1,s2);
    }
    else if(s1.size()==s2.size()){
        if(s1<s2){
            flag=1; //标记,结果为负输出负号
            swap(s1,s2);
        }
    }
}
```

```
a[0]=s1.size();
b[0]=s2.size();
for(int i=1;i<=a[0];i++)
    a[i]=s1[a[0]-i]-'0';
for(int i=1;i<=b[0];i++)
    b[i]=s2[b[0]-i]-'0';
Sub();
if(flag==1) cout<<"-";
for(int i=c[0];i>=1;i--)
    cout<<c[i];
cout<<endl;
return 0;
}
```



课堂练习



课堂练习

求斐波那契数列第 n 项的值。（大整数加法）
输入 n ，输出斐波那契数列第 n 项值。 $n \leq 500$ 。

参考代码

```
#include <iostream>
#include <cmath>
using namespace std;
string s[1005]={"0","1","1","2","3"};
string Plus(string s1,string s2)
{
    string ans;
    int a[100001],b[100001],c[100001];
    a[0]=s1.size();
    b[0]=s2.size();
    for(int i=1;i<=a[0];i++)
        a[i]=s1[a[0]-i]-'0';
    for(int i=1;i<=b[0];i++)
        b[i]=s2[b[0]-i]-'0';
    c[0]=max(a[0],b[0]);
    int jw=0,k;
```

参考代码

```
for(int i=1;i<=c[0];i++) //相加
{
    k=a[i]+b[i]+jw; //相加结果
    c[i]=k%10; //存进结果数组
    jw=k/10; //计算进位
}
if(jw!=0) //处理最高位的进位
{
    c[0]++;
    c[c[0]]=jw;
}
for(int i=c[0];i>=1;i--)
    ans+=char(c[i]+48);
return ans;
}
```

```
string Fib(int n)
{
    for(int i=5;i<=n;i++)
    {
        s[i]=Plus(s[i-1],s[i-2]);
    }
    return s[n];
}

int main()
{
    int n;
    cin>>n;
    cout<<Fib(n);
    return 0;
}
```

小结



1. 高精度加法思路是什么？
2. 高精度减法思路是什么？
3. 高精度加法和减法代码在写的时候有什么容易出错的地方？



作业

- 1、大整数加法 <http://noi.openjudge.cn/ch0106/10/>
- 2、大整数减法 <http://noi.openjudge.cn/ch0106/11/>