

- 模拟与高精度
 - 高精度加法
 - 核心代码
 - 例题
 - 高精度乘法
 - 核心代码
 - 例题

模拟与高精度

高精度加法

核心代码

*数据读取:要把最高位放在用来计算数组的最低位,也就是把输入的数字倒转再存(char -> int要补 '0')

```
for(int i=0;i<strlen(Arr1);i++)//arr用来计算, Arr用来读取
    arr1[i]=Arr1[strlen(Arr1)-i-1]-'0';
```

*高精度计算: 逐位计算, 注意进位

```
int temp=0//处理进位
int len=max(strlen(Arr1),strlen(Arr2))//操作次数
for(int i=0;i<len;i++)//主体,arr3[]存结果
{
    arr3[i]=arr1[i]+arr2[i]+temp;
    temp=arr3[i]/10;// '/'不能写成 '-'
    arr3[i]%=10;
}
```

*输出: 分最高位要不要进位两种情况讨论

```
if(temp!=0)//最高位还要进位的话需要多开一位空间
{
    arr3[len]=temp;
    for(int i=len;i>=0;i--)//倒着输出
        cout<<arr3[i];
}
```

```

else//否则直接倒序输出
{
    for(int i=len-1;i>=0;i--)//最高位len-1
        cout<<arr3[i];
}

```

例题

对于输入的两个不超过100位数字的非负整数，给出两数之和。

输入格式:

在两行中分别给出两个不超过100位数字的非负整数

输出格式:

在一行中输出两数之和

输入样例:

```

1 123
2 12

```

输出样例:

```

135

```

```

#include<bits/stdc++.h>
using namespace std;
char Arr1[1050],Arr2[1050];//因为要读取输入所以必须char类型 (int会报错)
int arr3[1050] ,arr1[1050],arr2[1050];
int main()
{
    cin>>Arr1>>Arr2;
    for(int i=0;i<strlen(Arr1);i++) arr1[i]=Arr1[strlen(Arr1)-i-1]-'0';
    for(int i=0;i<strlen(Arr2);i++) arr2[i]=Arr2[strlen(Arr2)-i-1]-'0';//将字符数组
    变成整数数组,然后倒转
    int temp=0; //用temp定义进位
    int len=max(strlen(Arr1),strlen(Arr2));//这里strlen不能用arr (类型不兼容)
    for(int i=0;i<len;i++)
    {
        arr3[i]=arr1[i]+arr2[i]+temp;
        temp=arr3[i]/10; //算出要不要进位 (注意不要用 temp=arr3[i]-10)
        arr3[i]=arr3[i]%10; //处理进位 (也要注意不要写成arr3[i]=arr3[i]-10)
    }
    if(temp!=0)
    {
        arr3[len]=temp;//如果最高位也要进位的话要多开一位空间
        for(int i=len;i>=0;--i)cout<<arr3[i];
        return 0;
    }
    for(int i=len-1;i>=0;--i)cout<<arr3[i];
}

```

```
        return 0;
    }
```

高精度乘法

核心代码

*读入数据并反转：

```
int len=s.size();//求计算次数
for(int i=0;i<len;i++)
    a[len-i]=s[i]-'0';
```

*高精度乘法（两层循环）：

```
for(int i=1;i<=len;i++)
{
    for(int j=1;j<=len;j++)
    {
        c[i+j-1]+=a[j]*b[i];
        c[i+j]+=c[i+j-1]/10;
        c[i+j-1]%=10;
    }
}
```

*删除前导0并输出

```
int lenc=2*len//乘完长度最大为2len
while(lenc>1&&c[lenc]==0) lenc--;//删除前导0
for(int i=lenc;i>=0;i--)
    cout<<c[i];
```

例题

T1.

高精度求累加和的问题与解

题目描述：

使用求和公式求1到N的累加和大家都会，但是如果把N值变大呢，比如100位的整数，那该怎么求？

输入格式：

输入在一行中给出1个位数不超过100位的整数N。

输出格式：

对每一组输入，在一行中输出1+2+3+.....+N的值。

输入样例：

在这里给出一组输入。例如：

```
1 | 10
```

输出样例：

在这里给出相应的输出。例如：

```
1 | 55
```

```
//用到高斯求和
#include<iostream>
#include<string.h>
using namespace std;
#define max 202
int a[max],b[max],c[max];
int main()
{
    string s;cin>>s;
    int x,i,j;
    int len=s.size();
    for(int i=0;i<len;i++){
        a[len-i]=s[i]-'0';//反转
        b[len-i]=s[i]-'0';
    }
    b[1]++;//后面用到高斯求和-- (1+n) *n*1/2,这里求的是 (1+n)
    for(int i=1;i<len;i++){//考虑进位 (可能连环进位)
        if(b[i]==10){
            b[i]=0;
            b[i+1]++;
        }
        else break;//只要有一个地方不用进位就可以berak了
    }
    for(int i=1;i<=len;i++){
        for(int j=1;j<=len;j++){
            c[i+j-1]+=a[j]*b[i];
            c[i+j]=c[i+j-1]/10;
            c[i+j-1]%=10;//高精度乘法 (体会一下)
        }
    }
    int lenc=2*len;//两个长度len的结果相乘最多长度变成2len
    for(int i=lenc;i>=1;i--){//把相乘结果÷2
```

```

        if(c[i]%2==0){
            c[i]/=2;
        }
        else{
            c[i-1]+=10;
            c[i]/=2;
        }
    }
    while(c[lenc]==0&&lenc>1)  lenc--; //去掉前导0
    for(int i=lenc;i>=1;i--) {
        cout<<c[i];
    }
}

```

T2.

7-2 a * b 分数 300

全屏浏览题目 切换布局

作者 neuqAcmClub

单位 东北大学秦皇岛分校

给出两个不超过1000位的十六进制数 a, b 。

求 $a * b$ 的值

输入格式:

输入共两行，两个十六进制的数

输出格式:

输出一行，表示 $a * b$

输入样例:

在这里给出一组输入。例如：

```
1BF52
1D4B42
```

输出样例:

在这里给出相应的输出。例如：

```
332FCA5924
```

代码长度限制	16 KB
时间限制	400 ms
内存限制	64 MB

```
#include<bits/stdc++.h>
using namespace std;
char a1[1000],b1[1000],c1[10000];
int a[1000],b[1000],c[100000];
int main()
{
    cin>>a1>>b1;
    int lena,lenb;
    lena=strlen(a1);
    lenb=strlen(b1);
    for(int i=1;i<=lena;i++)
```

```

    {
        if(a1[lena-i]>='A'&&a1[lena-i]<='Z')a[i]=a1[lena-i]-'7';//碰到字母用
        else a[i]=a1[lena-i]-'0';
    }
    for(int i=1;i<=lenb;i++)
    {
        if(b1[lenb-i]>='A'&&b1[lenb-i]<='Z')b[i]=b1[lenb-i]-'7';//碰到字母用
        else b[i]=b1[lenb-i]-'0';
    }
    int temp;
    for(int i=1;i<=lena;i++)
    {
        for(int j=1;j<=lenb;j++)
        {
            c[i+j-1]+=a[i]*b[j];
            c[i+j]+=c[i+j-1]/16;
            c[i+j-1]%=16;//高精度乘法(注意是十六进制所以除的是16)
        }
    }
    int lenc=lena+lenb;
    while(c[lenc]==0&&lenb>=1)lenc--;//删除前导0
    for(int i=1;i<=lenc;i++)
    {
        if(c[i]>=10&&c[i]<=15)c1[i]=c[i]+'7';
        else c1[i]=c[i]+'0';
    }
    for(int i=lenc;i>=1;i--)cout<<c1[i];
}

```