

2025-3-8 日 高精度乘法练习

补全程序练习题：（共三道题）

练习一：

```
/*
大数 A*int b
给出一个大数 A 和一个整数 b，两个数均大于等于 0。A 的长度小于等于 100000，整数
b 的值小于 10000。
输入数据
第 1 行：大数 A（长度小于等于 100000）（A>=0）
第 2 行：整数 b（ b>=0 && b<10000 ）
输出数据
输出 A*b 的结果
*/
#include<bits/stdc++.h>
using namespace std;
const int N=100010;//
int a[N];
//string 字符串按位转存到数组 a 中
void _save(string s,int a[]){
    int len=s.size();
    for(int i=0;i<len;i++){
        a[i]=s[len-1-i]-'0';
    }
}
int main(){
    string s;
    cin>>s;
    int b;
    cin>>b;
    _save(s,a);

    int lens=s.size();
    int lenb=0,b1=b;
    while(b1){
        b1=b1/10;
        lenb++;
    }
    int l=lens+lenb;
    for(int i=0;i<lens;i++) a[i]=a[i]*b;

    for(int i=0;i<l;i++){
        a[i+1]+=a[i]/10;
```

```

        a[i]=a[i]%10;
    }
    while(a[l-1]==0&&|>=1) l--;
    if(l==0) cout<<0;
    else
        for(int i=0;i<l;i++) cout<<a[l-1-i];
    return 0;
}

```

练习二：

/* 大数 A 乘以大数 B

给出 2 个大整数 A, B, 计算 A*B 的结果。

输入数据

第 1 行：大数 A

第 2 行：大数 A

(A,B 长度<=1000, A,B>=0)

输出数据

输出 A*B

*/

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

```
using namespace std;
```

```
const int N=1010,M=2010;
```

```
int a[N],b[N],c[M];
```

//自定义函数把 string 存储到 int 数组中

```
void save_1(string s,int a[]){
```

```
    //补全程序
```

```
    for(int i=0;i<s.size();i++){
```

```
        a[i]=s[s.size()-1-i]-'0';
```

```
    }
```

```
}
```

```
int main(){
```

```
    //定义 输入
```

```
    string s,s1;
```

```
    cin>>s>>s1;
```

```
    //存储 对齐
```

```
    save_1(s,a);
```

```
    save_1(s1,b);
```

```
    //模拟竖式 每位相乘 、进位 (补全程序)
```

```
    for(int i=0;i<s1.size();i++) { //循环把 b 各个位数拿出来
```

```
        for(int j=0;j<s.size();j++) c[j+i]+=a[j]*b[i];
```

```
        for(int j=0;j<s.size()+s1.size();j++){
```

```
            c[j+i+1]+=c[j+i]/10; //满 10 向前一位进 1
```

```
            c[j+i]=c[j+i]%10; //余数保存在本位上
```

```
        }
```

```

}
//输出
int l=s.size()+s1.size();
while(c[l-1]==0&& l>=1) l--;
if(l==0) cout<<0;
else for(int i=0;i<l;i++) cout<<c[l-1-i];
return 0;
}

```

练习三：

/*大数 A 乘以大数 B

给出 2 个大整数 A, B, 计算 A*B 的结果。

输入数据

第 1 行：大数 A

第 2 行：大数 A

(A,B 长度<=1000, A,B>=0)

输出数据

输出 A*B

(高精压位 10000 进制 代码实现)

*/

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

```
using namespace std;
```

```
const int N=1010;
```

```
int a[N],b[N],c[N+N];
```

//压位存储 进制 10000, 最大 9999, 四位数

```
void save_int(string s,int a[]){
```

```
    //补全程序 10000 进制存储
```

```
    int len=s.size();
```

```
    for(int i=0;i<len;i+=4){
```

```
        int k=1;
```

```
        for(int j=0;j<4&&j<=len-1-i;j++){
```

```
            a[i/4]+=(s[len-1-i-j]-'0')*k;
```

```
            k*=10;
```

```
        }
```

```
    }
```

```
}
```

```
void cal_s(int a[],int b[],int c[],int lena, int lenb, int lln){
```

```
    //完成两个大数相乘、进位
```

```
    // 循环依次取出 b[i]
```

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

```
        //b[i]依次与 a[i]的每一位相乘结果存在 c[i+j]中
```

```
        for(int j=0;j<lena;j++){
```

```
            c[i+j]+=b[i]*a[j];
```

```
        }
```

```

        //完成 c 的进位
        //补全程序
        for(int j=0;j<lln;j++){
            c[i+j+1]+=c[i+j]/10000;
            c[i+j]=c[i+j]%10000;
        }
    }
}

int main(){
    string s1,s2;
    cin>>s1>>s2;
    //字符串转存到数组中，完成压位
    save_int(s1,a);
    save_int(s2,b);
    //计算字符串长度
    int lena=s1.size()/4+1;
    int lenb=s2.size()/4+1;
    int lln=lena+lenb;
    //完成相乘、并把结果存储在数组 c[] 中
    cal_s(a,b,c,lena,lenb,lln);
    //反向输出结果
    while(c[lln-1]==0&&lln>=1)    lln--;
    if(lln==0){
        cout<<0;
        return 0;
    }
    //除了最高位，不足 4 位的在前面用 0 补齐
    for(int i=0;i<lln;i++){
        if(i==0) cout<<c[lln-1-i];
        else printf("%04d",c[lln-1-i]);
    }
    return 0;
}

```

编程题：（共 2 道题）

练习 1：ProjectEuler 20 求 n! 的各位和

$n!$ 表示 $n \times (n-1) \times \dots \times 3 \times 2 \times 1$ 比如 $10! = 10 \times 9 \times \dots \times 3 \times 2 \times 1 = 3628800$, $10!$ 的各位和是 $3 + 6 + 2 + 8 + 8 + 0 + 0 = 27$. 输入 n , 输出 $n!$ 的各位和。

输入数据

第 1 行：输入第一行组数 T , ($1 \leq T \leq 100$)
 接下来 T 行，每行一个整数 n . ($1 \leq N \leq 100$)

输入例：

3
3
6
100

输出数据

输出 T 行，每行输出一个数，表示 n! 的各位和

输出例：

6
9
648

参考程序：

```
#include<bits/stdc++.h>
using namespace std;
int c[1000];
void cal_ans(int n, int c[]){
    //数组 c[] 存储多个 n! 的结果，所以每次计算之前先初始化
    //memset(c,0,sizeof(c));//初始化数组各个位为 0
    c[0]=1;//初始化 c[0]=1，参与乘法计算
    //len 表示 c[] 参与计算的位数 ans 表示结果位数和
    int len=1,ans=0;
    for(int j=1;j<=n;j++){//计算 n!
        //j 和 c[] 各个位数相乘并把结果保存在 c[] 中
        for(int k=0;k<len;k++)    c[k]=c[k]*j;

        //计算 j 的位数
        int lenj=0,j_temp=j;
        while(j_temp){
            j_temp/=10;
            lenj++;
        }

        len=len+lenj;
        //完成进位 得到新的 c[]
        for(int k=0;k<len;k++){
            c[k+1]+=c[k]/10;
            c[k]=c[k]%10;
        }
    }
}
```

```

        //计算各个位数之和
        for(int k=0;k<=len;k++) ans+=c[k];
        //输出各个位数之和
        cout<<ans<<endl;
    }
    int main(){
        int t;
        cin>>t;
        for(int i=0;i<t;i++){//循环输入 t 个 n
            int n;
            cin>>n;
            memset(c,0,sizeof(c));//初始化数组各个位为 0
            //1.计算 n! 2.计算 n! 结果各个位数和 3.输出当前 n! 位数和
            cal_ans(n,c);
        }
        return 0;
    }
}

```

练习 2：阶乘之和

用高精度计算 $S=1! + 2! + 3! + \dots + n!$ ($n \leq 200$)

$n!$ 表示 $n \times (n-1) \times \dots \times 3 \times 2 \times 1$ 比如 $10! = 10 \times 9 \times \dots \times 3 \times 2 \times 1 = 3628800$,

输入数据

一个正整数 n

输入例：

6

输出数据

一个正整数 S ，表示计算结果

输出例：

873

参考程序：

```

#include<bits/stdc++.h>
using namespace std;
int c[1000],a[1000];
//大整数 A+B
void add_(int c[],int a[],int len){
    for(int k=0;k<len;k++){
        a[k]+=c[k];
    }
}

```

```

        for(int k=0;k<len+1;k++){
            a[k+1]+=a[k]/10;
            a[k]=a[k]%10;
        }
        while(c[len]==0&&len>=1) len--;
    }
    //大整数 A*int b
    int cal_ans(int n, int c[],int a[]){
        //数组 c[]存储多个 n! 的结果，所以每次计算之前先初始化
        c[0]=1;//初始化 c[0]=1，参与乘法计算
        //len 表示 c[]参与计算的位数
        int len=1;
        for(int j=1;j<=n;j++){//计算 n!
            //j 和 c[]各个位数相乘并把结果保存在 c[]中
            for(int k=0;k<len;k++)    c[k]=c[k]*j;

            //计算 j 的位数
            int lenj=0,j_temp=j;
            while(j_temp){
                j_temp/=10;
                lenj++;
            }
            len=len+lenj;
            //完成进位 得到新的 c[]
            for(int k=0;k<len;k++){
                c[k+1]+=c[k]/10;
                c[k]=c[k]%10;
            }
        }

        while(c[len-1]==0&&len>=1) len--;
        //调用函数 add_()把这步计算出来的 n! 累加在 a[]里
        //add_(c,a,len);
        return len;
    }

    int main(){
        int t,len;
        cin>>t;
        for(int i=1;i<=t;i++){//循环输入 t 个 n
            memset(c,0,sizeof(c));//初始化数组各个位为 0
            //1.计算 n!
            len=cal_ans(i,c,a);
            //2.把 n! 累加起来

```

```
        add_(c,a,len);
    }
    //输出阶乘累加和
    for(int k=0;k<len;k++){
        cout<<a[len-1-k];
    }
    return 0;
}
```