第 9 讲　高精度运算

整数基本类型的范围有效，最大的 long long 类型是 8 字节的，取值范围是-2^{63}~$2^{63}-1$，对于更大的数值就无能为力了。

long long: -9223372036854775808~ 9223372036854775807　(-2^{63}~$2^{63}-1$)

int: -2147483648~ 2147483647　(-2^{31}~$2^{31}-1$)

高精度计算的基本思想是用一个数组来保存一个大数，在此基础上进行数值计算。

两类问题：

1.参与运算的数是单精度，运算结果最后是高精度。

如求：2^{1000}的值。

2.参与运算的数本身就是高精度。

输入两个数 a 和 b，求 a+b 的和。0<a,b<=10^{200}。

处理方法：

位数（长度）放在 a[0]，把个位数存在 a[1]，十位数存在 a[2]，依次类推；

模拟手工计算的方法。

目录

1.6 编程基础之一维数组 1
12:计算 2 的 N 次方 1
10:大整数加法 2
11:大整数减法 3
14:求 10000 以内 n 的阶乘 4

1.6 编程基础之一维数组

12:计算 2 的 N 次方

描述

任意给定一个正整数 N(N<=100)，计算 2 的 n 次方的值。

输入

输入一个正整数 N。

输出

输出 2 的 N 次方的值。

样例输入

5

样例输出

32

尝试：

int：4 分

long long：7 分。%lld　%I64d

double:有效位数不够。

参考程序：

//12:计算 2 的 N 次方

```
#include<cstdio>
int a[40]={1,1};
int main(){
 int n;
 scanf("%d",&n);
 for(int i=1;i<=n;i++){
        for(int j=1;j<=a[0];j++) a[j]*=2;
        for(int j=1;j<=a[0];j++){
               a[j+1]+=a[j]/10;
               a[j]=a[j]%10;
        }
        if(a[a[0]+1]>0) a[0]++;
 }
 for(int i=a[0];i>=1;i--) printf("%d",a[i]);
 printf("\n");
 return 0;
}
```

10:大整数加法

描述

求两个不超过 200 位的非负整数的和。

输入

有两行，每行是一个不超过 200 位的非负整数，可能有多余的前导 0。

输出

一行，即相加后的结果。结果里不能有多余的前导 0，即如果结果是 342，那么就不能输出为 0342。

样例输入

```
22222222222222222222
33333333333333333333
```

样例输出

```
55555555555555555555
```

参考程序

```
#include<cstdio>
#include<cstring>
char s1[210],s2[210];
int a[210],b[210],c[210];
int main(){
   scanf("%s",s1);
   scanf("%s",s2);
   int l1=strlen(s1),l2=strlen(s2);
   for(int i=1;i<=l1;i++) a[i]=s1[l1-i]-48;
   for(int i=1;i<=l2;i++) b[i]=s2[l2-i]-48;
   int l=l1>l2?l1:l2;
   for(int i=1;i<=l;i++){
```

```
            a[i+1]+=(a[i]+b[i])/10;
            a[i]=(a[i]+b[i])%10;
        }
        l++;
        while(a[l]==0&&l>1) l--;
        for(int i=l;i>=1;i--) printf("%d",a[i]);
        printf("\n");
        return 0;
    }
```

11:大整数减法

描述

求两个大的正整数相减的差。

输入

共 2 行，第 1 行是被减数 a，第 2 行是减数 b(a > b)。每个大整数不超过 200 位，不会有多余的前导零。

输出

一行，即所求的差。

样例输入

```
99999999999999999999999999999999999999
9999999999999
```

样例输出

```
99999999999999999999999990000000000000
```

参考代码：

```
#include<cstdio>
#include<cstring>
char s1[210],s2[210];
int a[210],b[210],c[210];
int main(){
    scanf("%s",s1);
    scanf("%s",s2);
    int l1=strlen(s1),l2=strlen(s2);
    for(int i=1;i<=l1;i++) a[i]=s1[l1-i]-48;
    for(int i=1;i<=l2;i++) b[i]=s2[l2-i]-48;
    for(int i=1;i<=l1;i++){
            if(a[i]<b[i]){
                    a[i+1]--;
                    a[i]+=10;
            }
            a[i]-=b[i];
    }
```

```
    while(a[l1]==0&&l1>1) l1--;
    for(int i=l1;i>=1;i--) printf("%d",a[i]);
    printf("\n");
    return 0;
}
```

14:求 10000 以内 n 的阶乘

描述

求 10000 以内 n 的阶乘。

输入

只有一行输入，整数 n（0<=n<=10000）。

输出

一行，即 n!的值。

样例输入

100

样例输出

933262154439441526816992388562667004907159682643816214685929638952175999932299156089414639761565182862536979208272237582511852109168640000000000000000000000000

参考程序：

```
#include<cstdio>
int a[36000]={1,1};
int main(){
    int n;
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
            for(int j=1;j<=a[0];j++) a[j]*=i;
            for(int j=1;j<=a[0];j++){
                    a[j+1]+=a[j]/10;
                    a[j]=a[j]%10;
            }
            int k=a[0];
            int m=a[k+1];
            while(m>0) {
                    a[++k]=m%10;
                    m=m/10;
            }
            a[0]=k;
    }
    for(int i=a[0];i>=1;i--)
            printf("%d",a[i]);
    printf("\n");
    return 0;
}
```