学习目标

1. 学会高精度乘法的基本使用

2. 学会利用高精度乘法解决各种常见问题

知识讲授

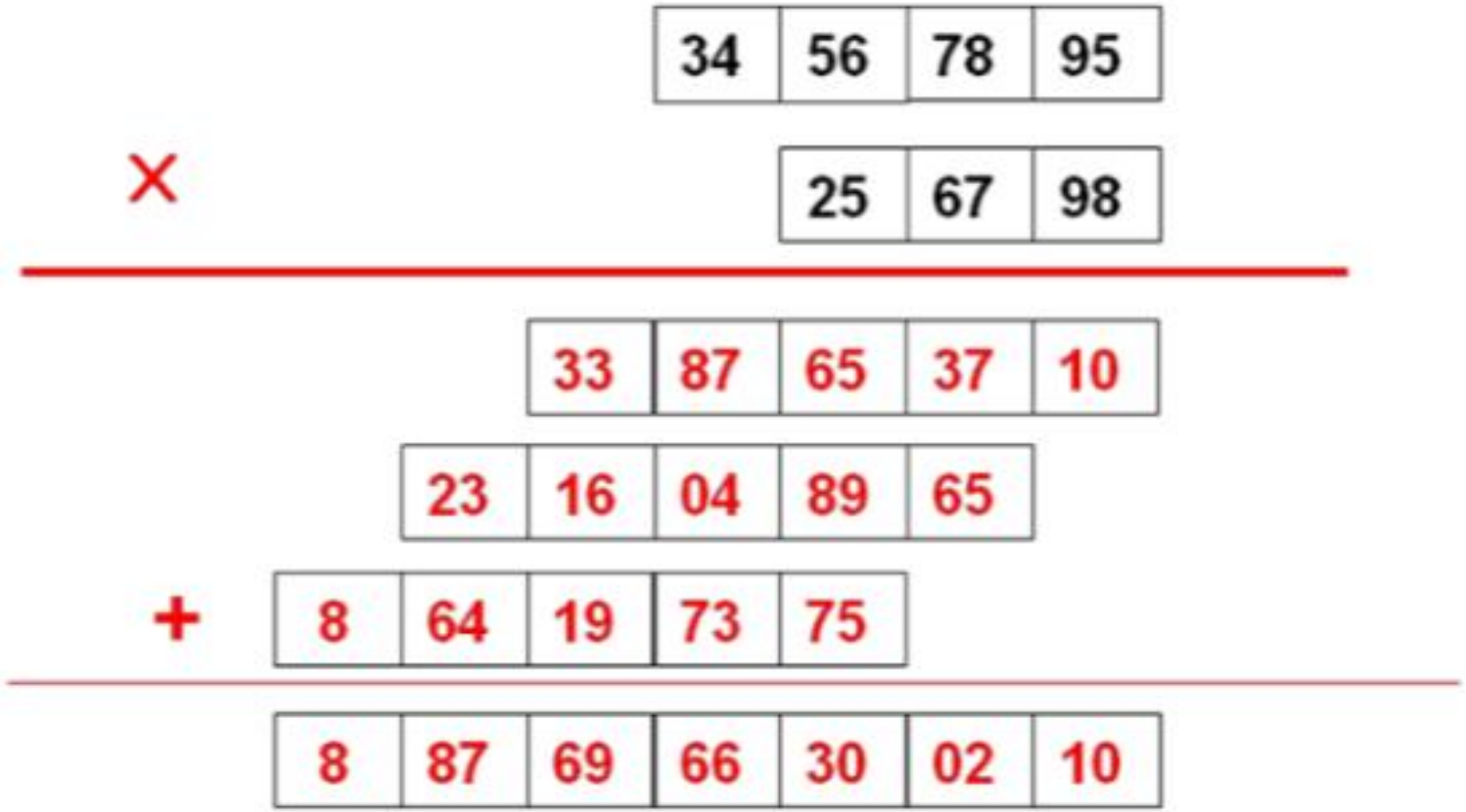
高精度乘法

7	6	5	4	3	2	1
ai				3	5	1
bi					6	2
				7	0	2
			21	0	6	
ci			21	7	6	2

高精度乘法

$$
\begin{array}{rccccccc}
 & a_n & \cdots\cdots & & & a_2 & a_1 \\
\times & b_n & \cdots\cdots & & & b_2 & b_1 \\
\hline
 & a_nb_1 & \cdots\cdots & & & a_2b_1 & a_1b_1 \\
a_nb_2 & a_{n-1}b_2 & \cdots\cdots & a_2b_2 & & a_1b_2 & \\
 & & \cdots\cdots & & & & \\
+ \quad a_nb_n & a_{n-1}b_n & \cdots\cdots & a_2b_n & a_1b_n & & \\
\hline
c_{2n} & c_{2n-1} & \cdots\cdots & c_n & \cdots\cdots & c_3 & c_2 & c_1
\end{array}
$$

高精度乘法-压位运算


34 56 78 95
×
25 67 98
33 87 65 37 10
23 16 04 89 65
+
8 64 19 73 75
8 87 69 66 30 02 10

高精度乘法

通过上图的竖式计算过程，我们可以总结出c数组下标的变化规律，可以写出如下关系式：ci = c’ i +c” i +...由此可见，c i 跟a[i]*b[j]乘积有关，还跟上次的进位有关，还跟原ci的值有关，于是可总结出如下规律：

```
c[i+j-1]+= a[i]*b[j]+ x + c[i+j-1];  //x为上一次进位。
```

高精度乘法示例

大数字乘法，两个非负数字相乘。

参考代码

```
#include <iostream>
#include <cmath>
using namespace std;

string s1,s2;
int a[100001],b[100001],c[100001];

void Mul()
{
   c[0]=a[0]+b[0];  //乘法结果最长位数就是两个位数相加
   int jw;
   for(int i=1;i<=b[0];i++) {
      jw=0;   //一轮循环开始，进位初始化为0；
      for(int j=1;j<=a[0];j++) {
         c[i+j-1]+=a[j]*b[i]+jw;
         jw=c[i+j-1]/10;   //计算进位
         c[i+j-1]%=10;     //存每位结果
      }
      c[i+a[0]] =jw;      //每一轮计算最高位的进位存储
   }
```

参考代码

```
    while(c[c[0]]==0&&c[0]>1)  //去前导0
        c[0]--;
}
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';
    Mul();
    for(int i=c[0];i>=1;i--)
        cout<<c[i];
    cout<<endl;
    return 0;
}
```

课堂练习

课堂练习

求5000以内数字的阶乘。（高精度×低精度）

参考代码

```
#include<iostream>
using namespace std;
int ans[1000005],jw,temp;
int main()
{
   int k=1,n; //k存结果的位数
   cin>>n;
   ans[1]=1;
   for(int i=1;i<=n;i++)
//枚举因式中的每一个因子
   {
      jw=0;
      for(int j=1;j<=k;j++) //把每一位数字都乘i
      {
         temp=ans[j]*i+jw;
         jw=temp/10;
         ans[j]=temp%10;
      }
      while(jw)
//把最后剩下的进位也拆出来存到数组里
      {
         ans[++k]=jw%10;
         jw/=10;
      }
   }
   for(int i=k;i>=1;i--)
      cout<<ans[i];
   cout<<endl;
   return 0;
}
```

课堂练习

编写一个程序来计算R^n的精确值，其中R是一个整数（0 < R <= 9999），n是一个整数（0 < n <= 250）。有多组数据输入。

参考代码

```
#include<iostream>
using namespace std;
int ans[100005],jw,temp,l;
int sum[100005];
void pw(int r,int n){
    int k=1; //K存结果的位数
    ans[1]=1; //初始化结果，ans求乘方
    for(int i=1;i<=n;i++){//枚举因式中的每一个因子
        jw=0;
        for(int j=1;j<=k;j++){ //把每一位数字都乘r
            temp=ans[j]*r+jw;
            jw=temp/10;
            ans[j]=temp%10;
        }
```

参考代码

```
    while(jw){  //把最后剩下的进位也拆出来存到数组里
          ans[++k]=jw%10;
          jw/=10;
       }
    }
    for(int i=k;i>=1;i--)cout<<ans[i];
    cout<<endl;
}
int main()
{
    int n,r,k;
    cin>>k;
    while(k--)  {
       cin>>r>>n;
       pw(r,n);
    }
    return 0;
}
```

小结

1. 高精度乘法基本思路是什么？

2. 计算结果时候数组之间有什么规律？

3. 如何计算阶乘的和？

作业

1、计算2的N次方 http://noi.openjudge.cn/ch0106/12/

2、求10000以内n的阶乘 http://noi.openjudge.cn/ch0106/14/

3、阶乘和 http://noi.openjudge.cn/ch0106/15/