

B MIPS 计算机 I

时间限制：2000ms 内存限制：65536kb

通过率：99/169 (58.58%) 正确率：99/870 (11.38%)

题目描述

你发现了一台无限位数的 MIPS 计算机！这台计算机的指令集中立即数均为十进制，并且系统调用会以十进制输出数字。

你有两个十进制非负整数 a, b ，并执行了以下 MIPS 汇编程序：

```
li $t1, <a>
li $t2, <b>
mul $a0, $t1, $t2
li $v0, 1
syscall
```

其中 <a> 指代十进制非负整数 a ， 指代十进制非负整数 b 。由于这台计算机位数无限，你不需要考虑溢出等问题。

为了验证 MIPS 计算机的正确性，你需要计算 MIPS 汇编程序输出的结果。

输入

一次测试会执行多次汇编程序。

第一行，一个正整数 T ($1 \leq T \leq 10$)，表示执行 MIPS 汇编程序的次数。

接下来 T 行，每行两个十进制非负整数 a, b ($0 \leq a, b < 10^{10^5}$)，表示向 MIPS 汇编程序中填充的 a, b 。

大数乘法
FFT
板子题

思路

对于每个 n 位的十进制数，看做一个 $n - 1$ 次多项式

$$A(x) = a_0 + a_1 \times 10 + a_2 \times 10^2 + \cdots + a_{n-1} \times 10^{n-1}$$

用 FFT 求出 $a \times b$ 的系数后，再模余进位即可

$$(0 \leq a, b < 10^{10^5})$$

2的18次方: 262144

注意如下易错点:

1. $a*b$ 位数上界大概有 $2e5$,但是由于FFT算法的特殊性(补全 2^n 次幂)数组应该要比这个开的大, 理论上开 $3e5$ 即可, 实际上我当时开了6次方(**)
2. 注意 $0 * 0$ 或者 $0 * \text{其他数}$ 的情况 是否考虑到了
3. 不建议使用递归(不仅速度慢, 且有REG的风险)
4. 很多同学REG的问题

```

const int ms=3e5+2;
const double PI=acos(-1);//arccos(1)=pai
int R[ms],ans[ms];
char s1[ms],s2[ms];
int T;
complex<double> a[ms],b[ms];//复数容器

```

```

1 void change(complex<double> A[],int n){
2     for(int i=0;i<n;i++){
3         R[i]=R[i/2]/2+(i%2)*(n/2);
4
5     for(int i=0;i<n;i++){
6         if(i<R[i])//保证每对数只翻转一次
7             swap(A[i],A[R[i]]);
8     }
9 }

```

```

void FFT(complex<double> A[],int n,int op){//op=1为正变换,op=-1为逆变换
    change(A,n);
    for(int m=2;m<=n;m<=1){
        complex<double> w1({cos(2*PI/m),sin(2*PI/m)*op});
        for(int i=0;i<n;i+=m){
            complex<double> wk({1,0});
            for(int j=0;j<m/2;j++){
                complex<double> x=A[i+j],y=A[i+j+m/2]*wk;
                A[i+j]=x+y;
                A[i+j+m/2]=x-y;
                wk=wk*w1;
            }
        }
    }
}

```

```

for(m+=n,n=1;n<=m;n*=2); //长度补成2的幂

FFT(a,n,1),FFT(b,n,1); //正变换
for(int i=0;i<n;i++)
    a[i]=a[i]*b[i];
FFT(a,n,-1); //逆变换

```

```

for(int i=0;i<n;i++){
    ans[i]+=(int)(a[i].real()/n+0.5);
    ans[i+1]+=ans[i]/10;
    ans[i]%=10;
}
while(n>=0&&!ans[n])
    n--;
if(n==-1)
    printf("0\n");
else{
    for(int i=n;i>=0;i--)
        printf("%d",ans[i]);
    printf("\n");
}

```

```

memset(ans,0,sizeof(ans));
for(int i=0;i<=p;i++)
    a[i].real(0),a[i].imag(0);
for(int i=0;i<=p;i++)
    b[i].real(0),b[i].imag(0);
memset(R,0,sizeof(R));

```

Thanks~