```
//FFT+高精度
typedef long long Long;
const int MAXN=32768;
const double pi=acos(-1.0);
const Long MOD=100000;
const int TEN=5;

double ra[MAXN];
double ia[MAXN];
double rb[MAXN];
double ib[MAXN];
double rc[MAXN];
double ic[MAXN];
char a[MAXN];
char b[MAXN];
int slena;
int slenb;
int lena;
int lenb;
int n,logn;
Long ans[MAXN];
double R[MAXN];
double I[MAXN];

int rev(int x,int bit)
{
      int ans=0;
      for (int i=0;i<bit;i++)
      {
            ans<<=1;
            if (x&1) ans|=1;
            x>>=1;
      }
      return ans;
}
```

```
void fft(double ir[],double ii[],int size,int mark)
{
      double delta=mark*2*pi;
      for (int i=0;i<size;i++)
      {
            int tt=rev(i,logn);
            R[tt]=ir[i];
            I[tt]=ii[i];
      }
      for (int s=1;s<=logn;s++)
      {
            int m=1<<s;
            double rwm=cos(delta/m);
            double iwm=sin(delta/m);
            for (int k=0;k<n;k+=m)
            {
                  double rw=1;
                  double iw=0;
                  for (int j=0;j<m/2;j++)
                  {
                        double rt=rw*R[k+j+m/2]-iw*I[k+j+m/2];
                        double it=rw*I[k+j+m/2]+iw*R[k+j+m/2];
                        double ru=R[k+j];
                        double iu=I[k+j];
                        R[k+j]=ru+rt;
                        I[k+j]=iu+it;
                        R[k+j+m/2]=ru-rt;
                        I[k+j+m/2]=iu-it;
                        double rnw=rw*rwm-iw*iwm;
                        double inw=rw*iwm+iw*rwm;
                        rw=rnw; iw=inw;
                  }
            }
      }
      for (int i=0;i<size;i++)
      {
            ir[i]=R[i];
            ii[i]=I[i];
      }
}
```

```
double
POW[10]={1,10,100,1000,10000,100000,1000000,10000000,100000000,1000000000};

int next(char str[])
{
      int len=0;
      for (str[len]=getchar();str[len]>='0';str[len]=getchar())
            len++;
      str[len]=0;
      return len;
}
int main()
{
      int nn=0;
      scanf("%d",&nn); getchar();
      while (nn--)
      {
            memset(ra,0,n<<3);
            memset(ia,0,n<<3);
            memset(rb,0,n<<3);
            memset(ib,0,n<<3);
            memset(ans,0,n<<3);

            slena=next(a);
            int cnt=0; lena=0;
            for (int j=slena-1;j>=0;j--)
            {
                  ra[lena]=ra[lena]+(a[j]-'0')*POW[cnt++];
                  if (cnt==TEN) {lena++; cnt=0;}
            }
            if (ra[lena]>0.1)    lena++;

            slenb=next(b);
            cnt=0; lenb=0;
            for (int j=slenb-1;j>=0;j--)
            {
                  rb[lenb]=rb[lenb]+(b[j]-'0')*POW[cnt++];
                  if (cnt==TEN) {lenb++; cnt=0;}
            }
            if (rb[lenb]>0.1)    lenb++;
```

```
        n=1; logn=0;
        while (n<lena || n<lenb) {n+=n;logn++;}
        n+=n; logn++;

        fft(ra,ia,n,1);
        fft(rb,ib,n,1);
        for (int i=0;i<n;i++)
        {
             rc[i]=ra[i]*rb[i]-ia[i]*ib[i];
             ic[i]=ra[i]*ib[i]+rb[i]*ia[i];
        }
        fft(rc,ic,n,-1);
        for (int i=0;i<n;i++)
             ans[i]=(Long)(rc[i]/n+0.5);
        for (int i=0;i<n-1;i++)
        {
             ans[i+1]+=ans[i]/MOD;
             ans[i]%=MOD;
        }
        bool print=0;
        for (int i=n-1;i>=0;i--)
        {
             if (!print && (ans[i]>0 || i==0))
             {
                  print=1;
                  printf("%lld",ans[i]);
             } else
             if (print)
                  printf("%05lld",ans[i]);
        }
        putchar(10);
    }
    return 0;
}
```