快速傅里叶（k进制）

×0需要特判 注意乘以后多项式长度会最大x2 在多组数据中记得将所有元素清零

调用流程：

```
fft(x1, len, 1);
fft(x2, len, 1);
for(int i = 0; i < len; i++)
    x1[i] = x1[i] * x2[i];
fft(x1, len, -1);
//此时x1实部为答案，注意此时答案并非k进制，还需要再进行进制转换
for(int i = 0; i < len; i++)
    sum[i] = (int)(x1[i].x + 0.5);
for(int i = 0; i < len; i++){
    sum[i + 1] += sum[i] / k;
    sum[i] %= k;
}
len = len1 + len2 - 1;
while(sum[len] <= 0 && len > 0) len--;
//从len到0是高位->低位
```

```
const double PI = acos(-1.0);
struct Complex
{
    double x,y;
    Complex(double _x = 0.0, double _y = 0.0){
        x = _x;
        y = _y;
    }
    Complex operator -(const Complex &b)const{
        return Complex(x - b.x, y - b.y);
    }
    Complex operator +(const Complex &b)const{
        return Complex(x + b.x, y + b.y);
    }
    Complex operator *(const Complex &b)const{
        return Complex(x * b.x - y * b.y, x * b.y + y * b.x);
    }
};
//改变进制
void change(Complex y[], int len){
    int i, j, k;
    for(i = 1, j = len / 2; i < len - 1; i++){
        if(i < j) swap(y[i], y[j]);
        k = len / 2;
        while(j >= k){
            j -= k;
```

```
            k /= 2;
        }
        if(j < k)
            j += k;
    }
}

void fft(Complex y[], int len, int on){
    change(y, len);
    for(int h = 2; h <= len; h <<= 1){
        Complex wn(cos(-on * 2 * PI / h), sin(-on * 2 * PI / h));
        for(int j = 0; j < len; j += h){
            Complex w(1, 0);
            for(int k = j; k < j + h / 2; k++){
                Complex u = y[k];
                Complex t = w * y[k + h / 2];
                y[k] = u + t;
                y[k + h / 2] = u - t;
                w = w * wn;
            }
        }
    }
    if(on == -1)
        for(int i = 0; i < len; i++)
            y[i].x /= len;
}

const int MAXN = 4000010;
Complex x1[MAXN], x2[MAXN];
int sum[MAXN];

int main(void)
{
    scanf("%d", &t);
    while(t--)
    {
        scanf("%d", &k);
        scanf("%s%s", str1, str2);
        int len1 = strlen(str1);
        int len2 = strlen(str2);
        if(str1[0] == '0' || str2[0] == '0')
        {
            printf("0\n");
            continue;
        }
        int len = 1;
        while(len < len1 * 2 || len < len2 * 2) len <<= 1;
        for(int i = 0; i < len1; i++)
            x1[i] = Complex(str1[len1 - 1 - i] - '0', 0);
        for(int i = len1; i < len; i++)
            x1[i] = Complex(0, 0);
        for(int i = 0; i < len2; i++)
            x2[i] = Complex(str2[len2 - 1 - i] - '0', 0);
        for(int i = len2; i < len; i++)
```

```
            x2[i] = Complex(0, 0);
        fft(x1, len, 1);
        fft(x2, len, 1);
        for(int i = 0; i < len; i++)
            x1[i] = x1[i] * x2[i];
        fft(x1, len, -1);
        for(int i = 0; i < len; i++)
            sum[i] = (int)(x1[i].x + 0.5);
        for(int i = 0; i < len; i++){
            sum[i + 1] += sum[i] / k;
            sum[i] %= k;
        }
        len = len1 + len2 - 1;
        while(sum[len] <= 0 && len > 0) len--;
        for(int i = len; i >= 0; i--)
                printf("%c", sum[i] + '0');
        printf("\n");
    }
    return 0;
}
```