因式分解

[CTSC2013] 因式分解

题目描述

通过代数基本定理，我们知道若计算重根，一个 n 次的多项式在复数域内恰好有 n 个零点(函数值为 0 的点)。现给定一个整系数多项式 F[x]，它的 n 个零点恰好都是有理数(即可以写成两个整数相除的形式)；同时，若我们把它所有的非零零点(函数自变量不为 0，函数值为 0)去重，则可以得到 r 个互不相同的非零零点，其中第 i 个非零零点可以被表示成下式：

$sgn_i \times \frac{q_i}{p_i}$

式中sgn_i表示第 i 个零点的符号，p_i和q_i为互质的两个正整数。

现在告诉你 F[x]，要求你输出将他因式分解后的形式。

输入格式

输入只有一行，包含多项式 F[x]。

多项式一定是如下的形式：

$a_n x\verb|^|n + a_{n-1} x\verb|^|n - 1 + \cdots a_1 x + a_0$

次数一定为从高到低，其中a_i为整数，并且若a_i为 0，则省略该项，若a_i为负数，则省略之前的加号，若a_i的绝对值为 1 且 i 不为 0，则不输出 1，并且保证a_n不为 0.

详见样例输入。

输出格式

输出一行，表示因式分解后的形式，格式如下：

$a_n(x + u_1/v_1)\verb|^|t_1(x + u_2/v_2)\verb|^|t_2 \ldots (x + u_s/v_s)\verb|^|t_s$

其中 u，v 互质，且 v 为正整数。

其中u_i/v_i从大到小排列，若$u_i/v_i = 0$ 则该项为 $x\verb|^|t_i$，若u_i/v_i为负数，则省略加号，若v_i为 1，则省略$/v_i$。

若t_i为 1 则省略$\verb|^|t_i$。

若a_n为± 1 则将 1 省略。

详见样例输出。

样例 #1

样例输入 #1

```
8x^7-258x^5+2112x^3-512x
```

样例输出 #1

```
8(x-4)^2(x-1/2)x(x+1/2)(x+4)^2
```

样例 #2

样例输入 #2

```
-x^2+2x-1
```

样例输出 #2

```
-(x-1)^2
```

提示

测试点编号	多项式最高次数	互异零点数	系数范围(绝对值)
1	2	2	≤ 10
2	4	4	≤ 100
3	7	7	$\le 10^6$
4	10	10	$\le 10^7$
5	12	12	$\le 10^{16}$
6	35	5	$\le 10^{24}$
7	39	5	$\le 10^{68}$
8	46	4	$\le 10^{104}$
9	80	2	$\le 10^{12}$
10	50	1	$\le 10^{316}$

p_i,q_i满足:

$\prod_{i=1}^{r}p_i\le 10^6,\prod_{i=1}^{r}q_i\le 10^6$

题解1

假设分解为$\Pi(p_ix+q_i)$，那么就有$\Pi p_i=A_n$，$\Pi q_i=A_0$。由于有$p_i<=1000000$，$q_i<=1000000$，我们枚举A_n和A_0的约数然后暴力枚举判断。。。

至于于是判断，可以考虑在模意义下判断，选70个大质数分别判断即可。。。

剩下的就是码码码了。。

AC代码如下：

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<algorithm>
#define ll long long
using namespace std;

const int prm[70]=
{543892411,567498259,643448581,772660877,802279559,823153213,847485637,907170331
,919507003,923425961,929812711,935151689,938910911,962946403,970870897,977946113
,985116131,989743967,991308817,994793521,995403061,997310857,998623607,999049111
,999860857,1000885111,1002002609,1002270383,1002326077,1002351107,1002642457,100
2681557,1003013519,1004475841,1005879367,1006478441,1007120461,1007468677,100757
6071,1008054149,1008640013,1008894637,1009316083,1009528721,1009862393,101051032
9,1010915173,1011460739,1012072651,1012612079,1012668131,1012825181,1014091867,1
014515053,1014904973,1015114333,1015944323,1016003431,1016121461,1017088367,1017
307463,1017423811,1017620873,1018394743,1018932493,1019199101,1019208581,1019319
187,1019376101,1019831203};
int n,m,cnt,tot1,tot2,b[70][85],p[100005],q[100005];
char s0[30005],s1[30005];
struct node{ int x,y,z; }ans[85];
bool cmp(const node &u,const node &v){
    return (ll)u.x*v.y>(ll)u.y*v.x;
}
struct hugnum{
    int len,sgn,num[40];
    void ins(int tot,int fu){
        int i,j; sgn=fu;
        for (i=tot; i>0; i-=9){
            len++;
            for (j=max(1,i-8); j<=i; j++) num[len]=num[len]*10+s1[j]-'0';
        }
    }
    int getmod(int x){
        int i,t=0;
        for (i=len; i; i--)
            t=((ll)t*1000000000+num[i])%x;
        return t;
    }
}a[85];
void get_in(){
    scanf("%s",s0+1);
```

```
    int i=1,len=strlen(s0+1);
    while (i<=len){
        int tot=0,fu=1,dgr=0;
        if (s0[i]=='-') fu=-1;
        if (s0[i]=='+' || s0[i]=='-') i++;
        if (s0[i]=='x'){
            s1[tot=1]='1'; i++;
            if (s0[i]=='^'){
                for (i++; s0[i]>='0' && s0[i]<='9'; i++)
                    dgr=dgr*10+s0[i]-'0';
                a[dgr].ins(tot,fu);
                if (!n) n=dgr;
            } else a[1].ins(tot,fu);
        } else{
            for (; s0[i]>='0' && s0[i]<='9'; i++) s1[++tot]=s0[i];
            if (!s0[i]) a[0].ins(tot,fu); else{
                i++;
                if (s0[i]=='^'){
                    for (i++; s0[i]>='0' && s0[i]<='9'; i++)
                        dgr=dgr*10+s0[i]-'0';
                    a[dgr].ins(tot,fu);
                    if (!n) n=dgr;
                } else a[1].ins(tot,fu);
            }
        }
    }
    if (a[n].sgn<0) putchar('-');
    if (a[n].len>1 || a[n].num[1]>1){
        printf("%d",a[n].num[a[n].len]);
        for (i=a[n].len-1; i; i--) printf("%09d",a[n].num[i]);
    }
}
void put_out(){
    sort(ans+1,ans+cnt+1,cmp); int i;
    for (i=1; i<=cnt; i++){
        if (!ans[i].x) putchar('x'); else
            if (ans[i].y==1)
                if (ans[i].x<0) printf("(x+%d)",-ans[i].x);
                else printf("(x%d)",-ans[i].x);
            else
                if (ans[i].x<0) printf("(x+%d/%d)",-ans[i].x,ans[i].y);
                else printf("(x%d/%d)",-ans[i].x,ans[i].y);
        if (ans[i].z>1) printf("^%d",ans[i].z);
    }
    puts("");
}
int ksm(int x,int y,int mod){
    int t=1; for (; y; y>>=1,x=(ll)x*x%mod) if (y&1) t=(ll)t*x%mod;
    return t;
}
int gcd(int x,int y){ return (y)?gcd(y,x%y):x; }
bool ok(int x,int y){
    int i,j,t,mod,tmp;
    for (i=0; i<70; i++){
        mod=prm[i]; t=(x+mod)%mod;
```

```
        t=(ll)t*ksm(y,mod-2,mod)%mod;
        for (j=n,tmp=0; j>=0; j--)
            tmp=((ll)tmp*t+b[i][j])%mod;
        if (tmp) return 0;
    }
    return 1;
}
void divide(int x,int y){
    int i,j,t,mod;
    for (i=0; i<70; i++){
        mod=prm[i]; t=(x+mod)%mod;
        t=(ll)t*ksm(y,mod-2,mod)%mod;
        for (j=n; j>=0; j--)
            b[i][j]=((ll)b[i][j+1]*t+b[i][j])%mod;
    }
}
int main(){
    get_in();
    int i=0,j,k; while (!a[i].len) i++;
    if (i>0) ans[++cnt]=(node){0,1,i};
    for (j=i; j<=n; j++) a[j-i]=a[j];
    n-=i;
    for (i=0; i<70; i++)
        for (j=0; j<=n; j++) b[i][j]=
(a[j].getmod(prm[i])*a[j].sgn+prm[i])%prm[i];
    for (i=1; i<=1000000; i++){
        if (!a[0].getmod(i)) p[++tot1]=i;
        if (!a[n].getmod(i)) q[++tot2]=i;
    }
    for (i=1; i<=tot1; i++)
        for (j=1; j<=tot2; j++) if (gcd(p[i],q[j])==1){
            for (k=0; ok(p[i],q[j]); k++)
                divide(p[i],q[j]);
            if (k) ans[++cnt]=(node){p[i],q[j],k};
            for (k=0; ok(-p[i],q[j]); k++)
                divide(-p[i],q[j]);
            if (k) ans[++cnt]=(node){-p[i],q[j],k};
        }
    put_out();
    return 0;
}
```

题解2

根据代数基本定理，当 $(x-a)$ 为 $f(x)$ 的因式时, $f(a)=0$ 。

设 $a_1,a_2,a_3,\ldots\ldots,a_i$ 为多项式 A 的根。

则 $A=(x-a_1)(x-a_2)\ldots\ldots(x-a_i)T$, T 为 $f(x)$ 在有理数范围内不可约为一次的因式。

举个例子:

设 $f(x)=x^3-3x^2-5x+14$ 。

若 $(x-a)$ 为 $f(x)$ 的因式,则 a 一定为 $\pm\frac{a_i}{b_i}$ ，其中 a_i 和 b_i 为最低次项和最高次项的因数。(数论知识，设为条件①)

当 $x=2$ 时，$f(x)=0$ ，因为 $2^3-3\times2^2-5\times2+14=0$ 。

可得 $f(x)=(x-2)\times(x^2-x+7)$ 。

给个证明:

若 $(a-x)$ 为 $f(x)$ 的因式，设 $g(x)\times(a-x)=f(x)$ 。

则 $f(a)=(a-a)\times g(a)=0$ ，得证。

在代码中我们每次将符合条件①的数带进去，枚举即可。

注意高精和约分为两大难点，代码写的足够详细了。

洛谷

```
#include<bits/stdc++.h>
using namespace std;
const int N=1000001;
//质数(素数)打表
const int su[75]=
{543892411,567498259,643448581,772660877,802279559,823153213,847485637,907170331
,919507003,923425961,929812711,935151689,938910911,962946403,970870897,977946113
,985116131,989743967,991308817,994793521,995403061,997310857,998623607,999049111
,999860857,1000885111,1002002609,1002270383,1002326077,1002351107,1002642457,100
2681557,1003013519,1004475841,1005879367,1006478441,1007120461,1007468677,100757
6071,1008054149,1008640013,1008894637,1009316083,1009528721,1009862393,101051032
9,1010915173,1011460739,1012072651,1012612079,1012668131,1012825181,1014091867,1
014515053,1014904973,1015114333,1015944323,1016003431,1016121461,1017088367,1017
307463,1017423811,1017620873,1018394743,1018932493,1019199101,1019208581,1019319
187,1019376101,1019831203};
int n,m,cnt,tot1,tot2,b[70][85],p[N],q[N];
char s1[N],s2[N];
struct nod{
    int fz,fm,cs;
}ans[85];//分数,名字写的很明白了
//fz分子fm分母cs次数
bool cmp(const nod &u,const nod &v){
    return (long long)u.fz*v.fm>(long long)u.fm*v.fz;
}//分数排序交叉相乘
struct gjd{//高精度
    int len,sgn,num[40];
    void tim(int ws,int fu){//压位乘
```

```
        int i,j; sgn=fu;
        for (i=ws; i>0; i-=9){
            len++;
            for (j=max(1,i-8); j<=i; j++) num[len]=num[len]*10+s2[j]-'0';
        }
    }
    int getmod(int x){//压位模
        int i,t=0;
        for (i=len; i; i--)
            t=((long long)t*1000000000+num[i])%x;
        return t;
    }
}a[85];
int ksm(int x,int y,int mod){
    int t=1;
    for (; y; y>>=1,x=(long long)x*x%mod) if (y&1) t=(long long)t*x%mod;
    return t;
}
int gcd(int x,int y){return (y)?gcd(y,x%y):x;}//最大公约数
bool pd(int x,int y){
    int i,j,t,mod,tmp;
    for (i=0; i<70; i++){
        mod=su[i]; t=(x+mod)%mod;
        t=(long long)t*ksm(y,mod-2,mod)%mod;
        for (j=n,tmp=0; j>=0; j--) tmp=((long long)tmp*t+b[i][j])%mod;
        if (tmp) return 0;
    }
    return 1;
}
void ys(int x,int y){
    int i,j,t,mod;
    for (i=0; i<70; i++){
        mod=su[i];
        t=(x+mod)%mod;
        t=(long long)t*ksm(y,mod-2,mod)%mod;
        for (j=n; j>=0; j--) b[i][j]=((long long)b[i][j+1]*t+b[i][j])%mod;
    }
}
int main(){
    ios_base::sync_with_stdio(false);
    scanf("%s",s1+1);//从1开始
    int i=1,len=strlen(s1+1);//strlen一定要加个一,否则会RE
    while (i<=len){
        int ws=0,fu=1,sh=0;
        if (s1[i]=='-') fu=-1;//负数
        if (s1[i]=='+' || s1[i]=='-') i++;//符号
        if (s1[i]=='x'){//'x'
            s2[ws=1]='1'; i++;
            if (s1[i]=='^'){//次方
                for (i++; s1[i]>='0' && s1[i]<='9'; i++)
                    sh=sh*10+s1[i]-'0';
                a[sh].tim(ws,fu);
                if (!n) n=sh;
            } else a[1].tim(ws,fu);
        } else{//系数
```

```
            for (; s1[i]>='0' && s1[i]<='9'; i++) s2[++ws]=s1[i];
            if (!s1[i]) a[0].tim(ws,fu); else{
                i++;
                if (s1[i]=='^'){
                    for (i++; s1[i]>='0' && s1[i]<='9'; i++) sh=sh*10+s1[i]-'0';
                    a[sh].tim(ws,fu);
                    if (!n) n=sh;
                }
                else a[1].tim(ws,fu);
            }
        }
    }
    if (a[n].sgn<0) putchar('-');//负数
    if (a[n].len>1 || a[n].num[1]>1){
        printf("%d",a[n].num[a[n].len]);
        for (i=a[n].len-1; i; i--) printf("%09d",a[n].num[i]);
    }
    //---分割线
    int j,k;
    i=0;//初始化
    while (!a[i].len) i++;
    if (i>0) ans[++cnt]=(nod){0,1,i};
    for (j=i; j<=n; j++) a[j-i]=a[j];
    n-=i;
    for (i=0; i<70; i++) for (j=0; j<=n; j++) b[i][j]=
(a[j].getmod(su[i])*a[j].sgn+su[i])%su[i];
    for (i=1; i<=1000000; i++){
        if (!a[0].getmod(i)) p[++tot1]=i;//首项因数
        if (!a[n].getmod(i)) q[++tot2]=i;//末项因数
    }
    for (i=1; i<=tot1; i++) for (j=1; j<=tot2; j++) if (gcd(p[i],q[j])==1){//互
质,减少次数
        for (k=0; pd(p[i],q[j]); k++) ys(p[i],q[j]);
        if (k) ans[++cnt]=(nod){p[i],q[j],k};//分数
        for (k=0; pd(-p[i],q[j]); k++) ys(-p[i],q[j]);
        if (k) ans[++cnt]=(nod){-p[i],q[j],k};//分数
    }
    //---分割线
    sort(ans+1,ans+cnt+1,cmp);
    for (i=1; i<=cnt; i++){
        if (!ans[i].fz) putchar('x');
        else if (ans[i].fm==1)
            if (ans[i].fz<0) printf("(x+%d)",-ans[i].fz);
            else printf("(x%d)",-ans[i].fz);
        else
            if (ans[i].fz<0) printf("(x+%d/%d)",-ans[i].fz,ans[i].fm);
            else printf("(x%d/%d)",-ans[i].fz,ans[i].fm);
        if (ans[i].cs>1) printf("^%d",ans[i].cs);
    }
    return 0;
}
```