Problem-1

```
#include<iostream>
using namespace std;
struct poly
{
    int coef;
    int expn;
    poly *next;
};
poly *h;
void CreatL(poly *h, int n)
{
    poly *p, *q;
    q = h;
    int c, e;
    for (int i = 0; i < n; i++)
    {
        cin >> c >> e;
        p = new poly;
        p->coef = c;
        p->expn = e;
        q->next = p;
        q = p;
    }
    q->next = NULL;
}
void ADD_1_2(poly *h1, poly *h2, poly *h3)
{
    poly *p, *q, *r, *s;
    p = h1->next;
    q = h2->next;
    s = h3;
    while (p&&q)
    {
        r = new poly;
        if (p->expn == q->expn)
        {
            r->coef = p->coef + q->coef;
            r->expn = p->expn;
            p = p->next;
            q = q->next;
        }
```

```
        else if (p->expn < q->expn)
        {

            r = p;
            p = p->next;
        }
        else
        {
            r = q;
            q = q->next;
        }
        s->next = r;
        s = r;
    }
    s->next = (p) ? p : q;
}
void PrintL(poly *h)
{
    poly *p;
    p = h->next;
    while (p)
    {
        if (p->coef != 0)
            cout << p->coef << " " << p->expn << endl;
        p = p->next;
    }
}
int main()
{
    int m, n;
    poly *h1, *h2, *h3;
    h1 = new poly;
    h2 = new poly;
    h3 = new poly;
    cin >> m;
    CreatL(h1, m);
    cin >> n;
    CreatL(h2, n);
    ADD_1_2(h1, h2, h3);
    PrintL(h3);
    return 0;
}
```

Problem-2

```
#include<iostream>
using namespace std;
struct poly
{
    int coef;
    int expn;
    poly *next;
};
poly *h;
void CreatLH(poly *h, int n)
{
    poly *p;
    int c, e;
    h->next = NULL;
    for (int i = n; i > 0; i--)
    {
        p = new poly;
        cin >> c >> e;
        p->coef = c;
        p->expn = e;
        p->next = h->next;
        h->next = p;
    }
}
void CreatLT(poly *h, int n)
{
    poly *p, *q;
    q = h;
    int c, e;
    for (int i = 0; i < n; i++)
    {
        cin >> c >> e;
        p = new poly;
        p->coef = c;
        p->expn = e;
        q->next = p;
        q = p;
    }
    q->next = NULL;
}
void Mul_1_2(poly *h1, poly *h2, poly *h3)
```

```
{
    poly *p, *q, *r;
    h3->next = NULL;
    p = h1->next;
    q = h2;
    while (q->next) q = q->next;
    int max_expn = p->expn + q->expn;
    for (int k = max_expn; k >= 0; k--)
    {
        int coe = 0;
        p = h1->next;
        while (p&&p->expn > k)
            p = p->next;
        q = h2->next;
        while (q&&p&&p->expn + q->expn < k)
            q = q->next;
        while (p&&q)
        {
            if (p->expn + q->expn == k)
            {
                coe += p->coef*q->coef;
                p = p->next;
                q = q->next;
            }
            else if (p->expn + q->expn < k)
                q = q->next;
            else
                p = p->next;
        }
        if (coe != 0)
        {
            r = new poly;
            r->coef = coe;
            r->expn = k;
            r->next = h3->next;
            h3->next = r;
        }
    }
}
void PrintL(poly *h)
{
    poly *p;
    p = h->next;
    while (p)
```

```
    {
        cout << p->coef << " " << p->expn << endl;
        p = p->next;
    }
}
int main()
{
    int m, n;
    poly *h1, *h2, *h3;
    h1 = new poly;
    h2 = new poly;
    h3 = new poly;
    cin >> m;
    CreatLH(h1, m);//h1头插法，降序输出（排列）
    cin >> n;
    CreatLT(h2, n);//h2尾插法，升序输出（排列）
    Mul_1_2(h1, h2, h3);//h3 插法，升序输出（排列）
    PrintL(h3);
    return 0;
}
```

Problem-3

```
#include<iostream>
#include<iomanip>
#include<cmath>
using namespace std;
struct poly
{
    int coef;
    int expn;
    poly *next;
};
void CreatL(poly *h, int n)
{
    poly *p, *q;
    q = h;
    int c, e;
    for (int i = 0; i < n; i++)
    {
        cin >> c >> e;
        p = new poly;
```

```
        p->coef = c;
        p->expn = e;
        q->next = p;
        q = p;
    }
    q->next = NULL;
}
void value_poly(poly *h, int x)
{
    poly *p;
    p = h->next;
    double value = 0;
    while (p)
    {
        value += p->coef* pow(x, p->expn);
        p = p->next;
    }
    cout << setiosflags(ios::fixed) << setprecision(1) << value <<
endl;
}
int main()
{
    int m;
    double x;
    poly *h;
    h = new poly;
    cin >> m;
    CreatL(h, m);
    cin >> x;
    value_poly(h, x);
    return 0;
}
```