

Προβολή κώδικα για το πρόβλημα 'polynomial'

<< Επιστροφή στο 'Εργαστήριο* progtech Σειρά #3 (για προπόνηση)'

Απλό κείμενο Αντιγραφή στο clipboard [Εκτύπωση](#) ?

```
01. #include <iostream>
02. #include <math.h>
03. using namespace std;
04.
05. class Polynomial{
06.     protected:
07.         class Term{
08.             protected:
09.                 int exponent;
10.                 int coefficient;
11.                 Term *next;
12.                 Term(int exp, int coeff, Term *n):exponent(exp),coefficient(coeff),next(n)
13.             {}
14.             friend class Polynomial;
15.             friend ostream & operator << (ostream &out, const Polynomial &p);
16.             friend Polynomial operator+ (const Polynomial &p, const Polynomial &q);
17.             friend Polynomial operator* (const Polynomial &p, const Polynomial &q);
18.         };
19.         Term *head;
20.     public:
21.         Polynomial(){
22.             head=nullptr;
23.         }
24.         Polynomial(const Polynomial &p){
25.             head=nullptr;
26.             Term *a=p.head;
27.             while(a!=nullptr){
28.                 addTerm(a->exponent,a->coefficient);
29.                 a=a->next;
30.             }
31.         }
32.         ~Polynomial(){
33.             Term *a=head;
34.             while(a!=nullptr){
35.                 a=a->next;
36.                 delete head;
37.                 head=a;
38.             }
39.         }
40.         Polynomial & operator = (const Polynomial &p){
41.             Term *a=head;
42.             while(a!=nullptr){
43.                 a=a->next;
44.                 delete head;
45.                 head=a;
46.             }
47.             head=nullptr;
48.             a=p.head;
49.             while(a!=nullptr){
50.                 addTerm(a->exponent,a->coefficient);
51.                 a=a->next;
52.             }
53.             return *this;
54.         }
55.     }
56.
57. void addTerm(int expon, int coeff){
58.     Term *p=head;
59.     Term *q=head;
60.     if(head==nullptr){
61.         head=new Term(expon,coeff,nullptr);
62.         return;
63.     }
64.     if(expon>head->exponent){
65.         head=new Term(expon,coeff,p);
66.         return;
67.     }
68.     else if(expon==head->exponent){
69.         head->coefficient+=coeff;
70.     }
```



Καλωσήρθες **pi17b350**

Όνομα: Account

Επίθετο: pi17b350

Έξοδος



Ενεργοί Διαγωνισμοί

- Προπόνηση progintro Σειρά #7
Απομένουν: 3696:29:08
- Εργαστήριο* progtech Σειρά #1 (για προπόνηση)
Απομένουν: 1464:29:08
- Εργαστήριο* progtech Σειρά #2 (για προπόνηση)
Απομένουν: 1464:29:08
- Εργαστήριο* progtech Σειρά #3 (για προπόνηση)
Απομένουν: 1464:29:08
- Εργαστήριο progtech Σειρά #4
Απομένουν: 288:29:08

```

71.         if(head->coefficient==0){
72.             if(head->next==nullptr)
73.                 head=nullptr;
74.             else
75.                 head=head->next;
76.             delete p;
77.         }
78.         return;
79.     }
80.     else{
81.         while(p->next!=nullptr){
82.             p=p->next;
83.             if(expon==p->exponent){
84.                 p->coefficient+=coeff;
85.                 if(p->coefficient==0){
86.                     q->next=p->next;
87.                     delete p;
88.                 }
89.                 return;
90.             }
91.             else if(expon>p->exponent){
92.                 q->next=new Term(expon,coeff,p);
93.                 return;
94.             }
95.             else{
96.                 q=q->next;
97.             }
98.         }
99.     }
100.     p->next=new Term(expon,coeff,nullptr);
101. }
102.
103. double evaluate(double x){
104.     double a=0;
105.     Term *p=head;
106.     while(p!=nullptr){
107.         a+=pow(x,p->exponent)*(p->coefficient);
108.         p=p->next;
109.     }
110.     return a;
111. }
112. Term * getHead()const{
113.     return head;
114. }
115. int absol(int x)const{
116.     return x>=0?x:-x;
117. }
118.
119. friend Polynomial operator+(const Polynomial &p, const Polynomial &q){
120.     Term *a=q.head;
121.     Polynomial summ;
122.     while(a!=nullptr){
123.         summ.addTerm(a->exponent,a->coefficient);
124.         a=a->next;
125.     }
126.     a=p.head;
127.     while(a!=nullptr){
128.         summ.addTerm(a->exponent,a->coefficient);
129.         a=a->next;
130.     }
131.     return summ;
132. }
133. friend Polynomial operator* (const Polynomial &p, const Polynomial &q){
134.     Term *a=p.head;
135.     Term *b=q.head;
136.     Polynomial prod;
137.     while(a!=nullptr){
138.         b=q.head;
139.         while(b!=nullptr){
140.             prod.addTerm((a->exponent)+(b->exponent),(a->coefficient)*(b-
141. >coefficient));
142.             b=b->next;
143.         }
144.         a=a->next;
145.     }
146.     return prod;
147. }
148.
149. friend ostream & operator <<(ostream &out,const Polynomial &q){
150.     Term *p=q.head;
151.     if(q.head==nullptr){
152.         out<<"0";
153.         return out;
154.     }
155.     char pros="+";
156.     if(p->coefficient<0){
157.         pros="-";
158.         out<<pros<<" ";
159.     }
160.     if(q.absol(p->coefficient)!=1)

```

```

160.         out<<q.absol(p->coefficient);
161.     if(p->exponent==1&&p->exponent!=0)
162.         out<<"x";
163.     else if(p->exponent!=0&&p->exponent!=1)
164.         out<<"x^"<<p->exponent;
165.     if(p->exponent==0&&q.absol(p->coefficient)==1)
166.         out<<"1";
167.     while(p->next!=nullptr){
168.         out<<" ";
169.         p=p->next;
170.         char pros='+';
171.         if(p->coefficient<0)
172.             pros='-';
173.         out<<pros<<" ";
174.         if(q.absol(p->coefficient)!=1)
175.             out<<q.absol(p->coefficient);
176.         if(p->exponent==1)
177.             out<<"x";
178.         if(p->exponent!=0&&p->exponent!=1)
179.             out<<"x^"<<p->exponent;
180.         if(p->exponent==0&&q.absol(p->coefficient)==1)
181.             out<<"1";
182.     }
183.     return out;
184. }
185.
186. };
187.
188. /*int main(){
189.     Polynomial p;
190.     p.addTerm(1,3);
191.     p.addTerm(2,1);
192.     p.addTerm(0,-1);
193.     Polynomial q(p);
194.     q.addTerm(1,-3);
195.
196.     cout<<p<<endl<<p.evaluate(1)<<endl<<q<<endl<<q.evaluate(1)
197. <<endl<<p+q<<endl<<p*q<<endl;
198.     return 0;
199. }*/

```


