

[DSLK]. Bài 8. DSLK Đa thức

Cho 2 đa thức $A(x)$ và $B(x)$, tiến hành tính tổng 2 đa thức này và in ra theo thứ tự bậc giảm dần. Sử dụng một node trong DSLK như sau :

```
4
5 □ struct node{
6     int bac;
7     int heso;
8     struct node *next;
9 }
10
```

Ví dụ $A(x) = 3x^2 + 5x + 7$, $B(x) = x^4 + 3x + 2$ thì bạn phải in ra đa thức tổng là $x^4 + 3x^2 + 3x + 9$.

```
#include <bits/stdc++.h>

using namespace std;

struct node{
    int heso, somu;
    node *next;
};

node* makeNode(int heso, int somu){
    //cap phat dong
    node *ptr = new node;
    //gan du lieu
    ptr->heso = heso;
    ptr->somu = somu;
    ptr->next = NULL;
    return ptr;
}

void duyet(node *head){
    while(head != NULL){
        //truy cap du lieu cua node head
        cout << head->heso << "x^" << head->somu;
        //tu node hien => nhay ra node dung sau
        head = head->next; // i++
        if(head != NULL) cout << " + ";
    }
    cout << endl;
}

void them(node *&head, int hs, int sm){
    node *temp = head;
    node *prev = head;
    while(temp != NULL){
        if(temp->somu == sm){
            temp->heso += hs;
            return;
        }
        prev = temp;
        temp = temp->next;
    }
    node *newNode = makeNode(hs, sm);
    prev->next = newNode;
}
```

```

    }
    node *newNode = makeNode(hs, sm);
    if(prev == NULL){
        head = newNode; return;
    }
    prev->next = newNode;
}

void sx(node *&head){
    for(node *i = head; i != NULL; i = i->next){
        node *min = i;
        for(node *j = i->next; j != NULL; j = j->next){
            if(min->somu < j->somu){
                min = j;
            }
        }
        swap(min->somu, i->somu);
        swap(min->heso, i->heso);
    }
}

int main(){
    node *head = NULL;
    string s;
    while(cin >> s){
        if(s != "+"){
            int hs = 0, sm = 0;
            int i = 0;
            while(s[i] != 'x'){
                hs = hs * 10 + (s[i] - '0');
                ++i;
            }
            i += 2;
            while(i < s.size()){
                sm = sm * 10 + (s[i] - '0');
                ++i;
            }
            them(head, hs, sm);
        }
    }
    sx(head);
    duyet(head);
}

```

Input Format

Dòng 1 là đa thức A(x).

Dòng 2 là đa thức B(x).

Constraints

Mỗi đa thức có bậc không quá 50, số lượng kí tự trong 1 đa thức không quá 10000 kí tự.

Output Format

In ra đa thức tổng của 2 đa thức A, B theo thứ tự giảm dần về bậc của biến.

Sample Input 0

```

1x^0 + 4x^3 + 5x^2 + 5x^1 + 1x^0 + 1x^1 + 3x^2 + 3x^3 + 1x^1
2x^0 + 5x^4 + 4x^0 + 4x^1 + 1x^0 + 2x^0 + 1x^3 + 3x^2 + 3x^3

```

Sample Output 0

$$5x^4 + 11x^3 + 11x^2 + 11x^1 + 11x^0$$