**Assignment Q1(1201)**

#include <iostream>

using namespace std;

class node

{

public:

    int co;

    int po;

    node \*next;

};

void sortlist(node \*u)

{

    node \*temp = NULL;

    int s1, s2;

    if (u != NULL)

    {

        while (u != NULL)

        {

            temp = u->next;

            while (temp != NULL)

            {

                if (temp->po > u->po)

                {

                    s1 = u->co;

                    s2 = u->po;

                    u->co = temp->co;

                    u->po = temp->po;

                    temp->co = s1;

                    temp->po = s2;

                }

                temp = temp->next;

            }

            u = u->next;

        }

    }

}

void get(node \*p)

{

    int coefficient, power, cont, u;

    cout << "how many terms :";

    cin >> u;

    cout << "Enter coefficient and power (4x3 = 4 3) : ";

    for (int i = 1; i <= u; i++)

    {

        node \*n = new node();

        n->next = NULL;

        cin >> coefficient >> power;

        p->co = coefficient;

        p->po = power;

        p->next = n;

        p = n;

    }

}

void print(node \*h)

{

    while (h->next != NULL)

    {

        cout << h->co << "x^" << h->po;

        h = h->next;

        if (h->next != NULL)

            cout << "+";

    }

}

void add(node \*p, node \*p1, node \*p2)

{

    p->next = NULL;

    while (p1 && p2)

    {

        node \*w = new node();

        if (p1->po > p2->po)

        {

            p->co = p1->co;

            p->po = p1->po;

            p1 = p1->next;

        }

        else if (p1->po < p2->po)

        {

            p->co = p2->co;

            p->po = p2->po;

            p2 = p2->next;

        }

        else

        {

            p->co = p1->co + p2->co;

            p->po = p1->po;

            p1 = p1->next;

            p2 = p2->next;

        }

        if (p1 && p2)

        {

            p->next = w;

            p = w;

            p->next = NULL;

        }

    }

    while (p1 || p2)

    {

        node \*ww = new node();

        p->next = ww;

        p = ww;

        p->next = NULL;

        if (p1)

        {

            p->co = p1->co;

            p->po = p1->po;

            p1 = p1->next;

        }

        else if (p2)

        {

            p->co = p2->co;

            p->po = p2->po;

            p2 = p2->next;

        }

    }

}

int main()

{

    node \*first = new node();

    node \*second = new node();

    node \*result = new node();

    printf("\nEnter the corresponding data:-\n");

    printf("\nFirst polynomial:\n");

    get(first);

    sortlist(first);

    print(first);

    printf("\nSecond polynomial:\n");

    get(second);

    sortlist(second);

    print(second);

    cout << endl;

    add(result, first, second);

    print(result);

    return 0;

}

**Assignment Q2(1201)**

#include <iostream>

using namespace std;

class node

{

public:

    int data;

    node \*left, \*right;

};

node \*newNode(int data)

{

    node \*temp = new node();

    temp->data = data;

    temp->left = temp->right = NULL;

    return temp;

}

void lsum(node \*r, int &sum)

{

    if (!r)

    {

        return;

    }

    if (!r->left && !r->right)

    {

        sum += r->data;

    }

    lsum(r->left, sum);

    lsum(r->right, sum);

}

int main()

{

    node \*root = newNode(2);

    root->left = newNode(7);

    root->left->left = newNode(2);

    root->left->right = newNode(6);

    root->left->right->right = newNode(11);

    root->left->right->left = newNode(5);

    root->right = newNode(5);

    root->right->right = newNode(9);

    root->right->right->left = newNode(4);

    int sum = 0;

    lsum(root, sum);

    cout << "Sum of the Leaf nodes : " << sum << endl;}