

PROGRAM TITLE:Add two Polynomials.

THEORY:A n^{th} polynomial is written as

$$a_0x^n + a_1x^{n-1} + \dots + a_n=0 \quad ; \text{on condition } a_0 \neq 0$$

where a_0, a_1, \dots, a_n are called the coefficients and the powers of x is it's degree.

PROGRAM ALGORITHM:

Algo_sortedinput (start)

```
{
    while(user wants to input more terms)
    {
        ask user for coefficient and degree of individual term;
        if(coefficient not equal to zero)
        {
            create term with inputs;
            insert term into polynomial in it's correct position so
as the whole polynomial stays in descending order;
        }
    }
    return;
}
```

Algo_polyadd(start1,start2,startfinal) //start1 implies the 1st term, start2 the 2nd term and startfinal where to store the result.

```
{
    p=start1;
    q=start2;
    r=startfinal;
    while(p and q both not equal to NULL)
    {
        if(deg(p)>deg(q)) //deg refers to degree of the term
        {
            add contents of p to r;
            p=next(p);
        }
        else if(deg(p)<deg(q))
        {
            add contents of q to r;
            q=next(q);
        }
        else
        {
            if(resultant coefficient not equal to zero)
                add resultant term to r;
            p=next(p);
            q=next(q);
        }
        if(r=NULL)
            initialise startfinal;
        else
            r=next(r);
    }
}
```

```

    }
    while(p not equal to NULL)
    {
        add term to r;
        p=next(p);
        r=next(r);
    }
    while(q not equal to NULL)
    {
        add term to r;
        q=next(q);
        r=next(r);
    }
    return;
}

```

PROGRAM CODE:

```

/*C++ Program to add two polynomials with the help of operator
overloading*/
#include <iostream>
#include <cstdlib>
using namespace std;

struct Term
{
    int co,deg;
    struct Term *next;
};
typedef struct Term *POLY;
class Poly
{
    POLY start;
public:
    Poly();
    POLY allocate_term(int,int);
    void input();
    Poly operator+(Poly);
    void print();
};
Poly::Poly()
{
    start=NULL;
}

/*Allocates memory space for a new term*/
POLY Poly::allocate_term(int co,int deg)
{
    POLY temp = (POLY)malloc(sizeof(struct Term));
    temp->co=co;
    temp->deg=deg;
    temp->next=NULL;
    return temp;
}

```

```

/*To take input of the polynomial*/
void Poly::input()
{
    char ch='Y';
    int i=1,co,de;
    while(ch=='Y' || ch=='y')
    {
        cout<<"\n\tEnter term no "<<i++<<" ";
        cout<<"\n\tEnter coefficient::";
        cin>>co;
        cout<<"\tEnter degree::";
        cin>>de;
        if(co!=0)
        {
            POLY temp=allocate_term(co,de);
            if(start==NULL)
            {
                start=temp;
            }
            else
            {
                POLY p=start,q;
                while( (p!=NULL) && (temp->deg) < (p->deg) )
                {
                    q=p;
                    p=p->next;
                }

                /*Adds the coefficients if the degrees are same*/
                if( (p!=NULL) && (temp->deg==p->deg) )
                    p->co=p->co+temp->co;
                else
                {
                    temp->next=p;
                    if(p==start)
                        start=temp;
                    else
                        q->next=temp;
                }
            }
        }
        cout<<"\n\tDo you want to give another term? (Y/N)::";
        cin>>ch;
    }
    print();
}

/*Adds the polynomials together*/
Poly Poly::operator+(Poly a)
{
    POLY p=start,q=a.start,temp;
    Poly ob;
    POLY r=ob.start;
    while(p!=NULL&&q!=NULL)
    {

```

```

        if ((p->deg) > (q->deg))
        {
            temp=allocate_term(p->co,p->deg);
            p=p->next;
        }
        else if ((p->deg) < (q->deg))
        {
            temp=allocate_term(q->co,q->deg);
            q=q->next;
        }
        else
        {
            if (p->co+q->co!=0)
                temp=allocate_term((p->co)+(q->co),p->deg);
            p=p->next;
            q=q->next;
        }
        if (r==NULL)
        {
            ob.start=r=temp;
        }
        else
        {
            r->next=temp;
            r=r->next;
        }
    }
    while (p!=NULL)
    {
        temp=allocate_term(p->co,p->deg);
        p=p->next;
        r->next=temp;
        r=r->next;
    }
    while (q!=NULL)
    {
        temp=allocate_term(q->co,q->deg);
        q=q->next;
        r->next=temp;
        r=r->next;
    }
    return ob;
}

/*prints the whole polynomial*/
void Poly::print()
{
    POLY p=start;
    if (p==NULL)
    {
        cout<<"\n\tThe polynomial doesnt exist";
    }
    else
    {
        cout<<"\n\tThe polynomial as of now is::\n\t";
    }
}

```

```

        while(p!=NULL)
        {
            if(p->co!=1)
            {
                if(p->deg==0)
                    cout<<p->co;
                else
                    cout<<(p->co)<<"x^"<<(p->deg);
            }
            else
                cout<<"x^"<<(p->deg);
            if((p->next!=NULL) && (p->next->co)>0)
                cout<<" ";
            p=p->next;
        }
        cout<<"\n";
    }
}

int main()
{
    Poly a,b,c;
    system("clear");
    cout<<"\n\tEnter 1st polynomial:-";
    a.input();
    cout<<"\n\tEnter 2nd polynomial:-";
    b.input();
    c=a+b;
    cout<<"\n\tThe First polynomial is::";
    a.print();
    cout<<"\n\tThe Second polynomial is::";
    b.print();
    cout<<"\n\tThe Final polynomial is::";
    c.print();
    return 0;
}

```

OUTPUT:

```

Enter 1st polynomial:-
Enter term no 1:
Enter coefficient::200
Enter degree::200

Do you want to give another term? (Y/N)::y

Enter term no 2:
Enter coefficient::-1
Enter degree::100

Do you want to give another term? (Y/N)::y

Enter term no 3:
Enter coefficient::5
Enter degree::93

```

Do you want to give another term? (Y/N)::y

Enter term no 4:

Enter coefficient::-5

Enter degree::87

Do you want to give another term? (Y/N)::y

Enter term no 5:

Enter coefficient::-6

Enter degree::90

Do you want to give another term? (Y/N)::n

The polynomial as of now is::

$200x^{200} - 1x^{100} + 5x^{93} - 6x^{90} - 5x^{87}$

Enter 2nd polynomial:-

Enter term no 1:

Enter coefficient::5

Enter degree::80

Do you want to give another term? (Y/N)::y

Enter term no 2:

Enter coefficient::1

Enter degree::99

Do you want to give another term? (Y/N)::y

Enter term no 3:

Enter coefficient::1

Enter degree::100

Do you want to give another term? (Y/N)::y

Enter term no 4:

Enter coefficient::6

Enter degree::87

Do you want to give another term? (Y/N)::n

The polynomial as of now is::

$x^{100} + x^{99} + 6x^{87} + 5x^{80}$

The First polynomial is::

The polynomial as of now is::

$200x^{200} - 1x^{100} + 5x^{93} - 6x^{90} - 5x^{87}$

The Second polynomial is::

The polynomial as of now is::

$x^{100} + x^{99} + 6x^{87} + 5x^{80}$

The Final polynomial is::

The polynomial as of now is::

$200x^{200} + x^{99} + 5x^{93} - 6x^{90} + x^{87} + 5x^{80}$

DISCUSSION:

- 1.The complexity of adding two polynomials is $O(n)$.
- 2.The program does not store the term if the coefficient entered is 0.
- 3.Provision has been made in the program for the case where the user inputs two polynomials of the same degree.