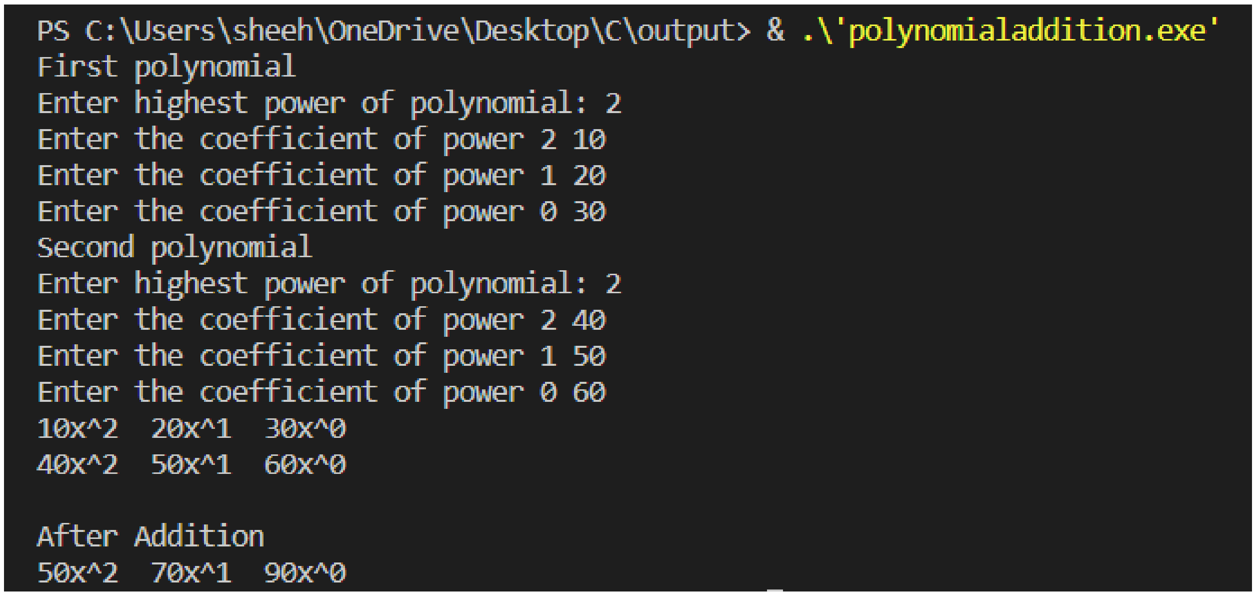
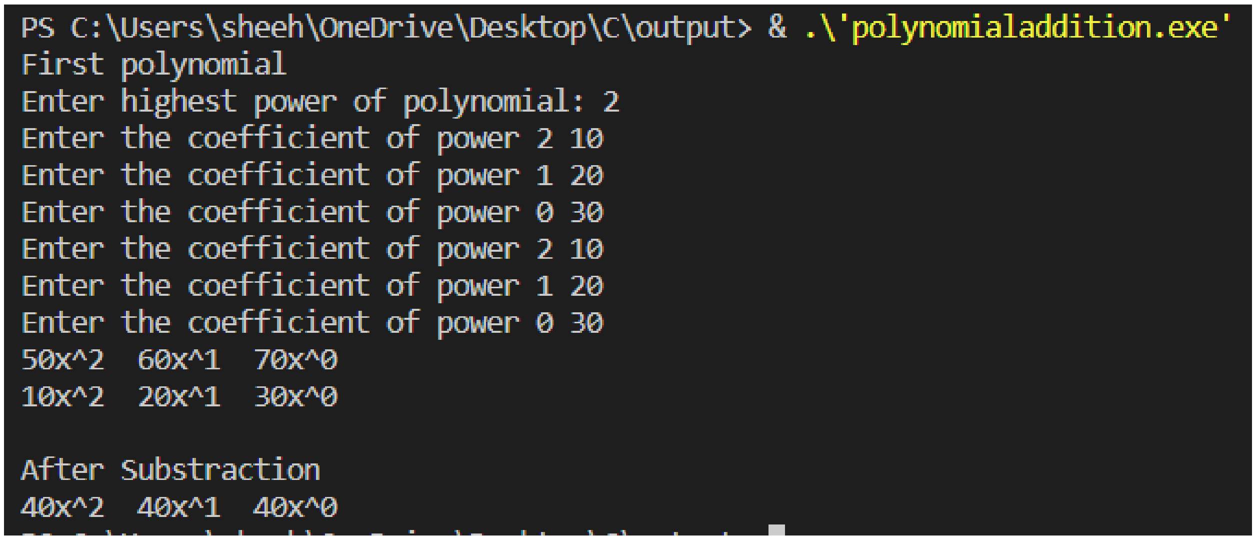
Name – Shrinivas Hatyalikar
Div – CS-B
Roll no – 24

Polynomial Addition,Subtraction and Multiplication

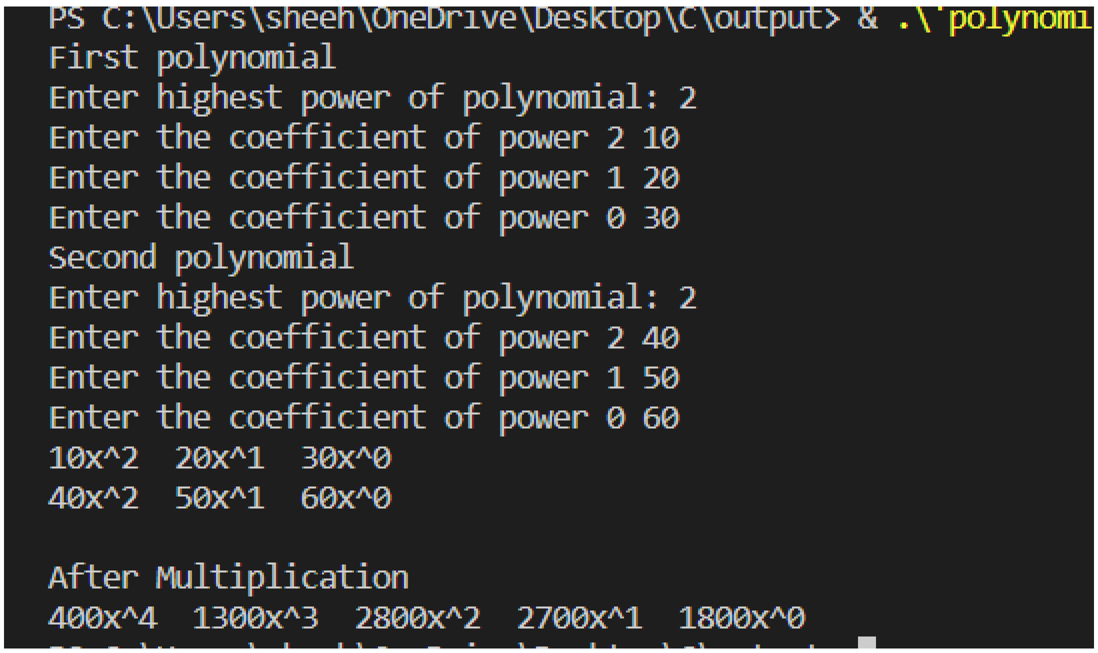
```
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    int pow;
    struct node *next;
}*head1,*head2,*result;

struct node *create(){
    struct node *head = (struct node *)malloc(sizeof(struct node));

    return (head);
}

void insert(struct node *head){
    int hig_pow,x;
    struct node *ptr;
    printf("Enter highest power of polynomial: ");
    scanf("%d",&hig_pow);
    printf("Enter the coefficient of power %d ",hig_pow);
    scanf("%d",&x);
    head->data=x;
    head->pow=hig_pow;
    head->next=NULL;
    ptr=head;
    for(int i=hig_pow-1;i>=0;i--){
        ptr->next=(struct node *)malloc(sizeof(struct node));
        printf("Enter the coefficient of power %d ",i);
        scanf("%d",&x);
        ptr=ptr->next;
        ptr->pow=i;
        ptr->data=x;
        ptr->next=NULL;
    }
}

void printlist(struct node *head)
{
    struct node *ptr=head;
    while (ptr!= NULL)
    {
        printf("%dx^%d  ",ptr->data, ptr->pow);
        ptr = ptr->next;
    }
}

struct node *add(struct node *head1,struct node *head2){
    struct node *p;
    if(head1!=NULL && head2==NULL){
        return head1;
    }
    if(head2!=NULL && head1==NULL){
        return head2;
    }
    while(head1!=NULL && head2!=NULL){
        if(result == NULL){
            result = (struct node *)malloc(sizeof(struct node));
            p=result;
        }
        else{
            p->next=(struct node *)malloc(sizeof(struct node));
            p=p->next;
        }
        if(head1->pow == head2->pow){
            //p->data=head1->data + head2->data;
            p->data=head1->data - head2->data;
            p->pow=head1->pow;
            head1=head1->next;
            head2=head2->next;
        }

        else if(head1->pow > head2->pow){
            p->data=head1->data;
            p->pow=head1->pow;
            head1=head1->next;

        }
        else if(head1->pow < head2->pow){
            p->data=head2->data;
            p->pow=head2->pow;
            head2=head2->next;
        }

    }
    while(head1!=NULL){
        p->next=(struct node *)malloc(sizeof(struct node));
        p=p->next;
        p->data=head1->data;
        p->pow=head1->pow;
        head1=head1->next;
    }
    while(head2!=NULL){
        p->next=(struct node *)malloc(sizeof(struct node));
        p=p->next;
        p->data=head2->data;
        p->pow=head2->pow;
        head2=head2->next;
    }
    p->next=NULL;
    return result;
}

void duplicate(struct node *head)
{
    struct node *p = head, *q, *del;
    while (p && p->next)
    {
        q = p;
        while (q->next)
        {
            if (p->pow == q->next->pow)
            {
                p->data = p->data + q->next->data;
                del = q->next;
                q->next = q->next->next;
                free(del);
                if (q)
                {
                    q = q->next;
                }
            }
            else
            {
                q = q->next;
            }
        }
        p = p->next;
    }
}

struct node *multiply(struct node *head1, struct node *head2)
{
    struct node *p = head1, *q = head2, *result1, *res2 = create();
    result1 = res2;
    while (p)
    {
        while (q)
        {
            result1->data = p->data * q->data;
            result1->pow = p->pow + q->pow;
            result1->next = NULL;
            q = q->next;
            if (q)
            {
                result1->next = (struct node *)malloc(sizeof(struct node));
                result1 = result1->next;
            }
        }
        q = head2;
        p = p->next;
        if (p)
        {
            result1->next = (struct node *)malloc(sizeof(struct node));
            result1 = result1->next;
        }
    }
    // print(res2);
    duplicate(res2);

    return res2;
}
int main(){
    struct node *p;
    printf("First polynomial\n");
    head1=create();
    insert(head1);
    printf("Second polynomial\n");
    head2=create();
    insert(head2);
    printlist(head1);
    printf("\n");
    printlist(head2);
    printf("\n");
    p=add(head1,head2);
    printf("\nAfter Addition\n");
    printf("\nAfter Substraction\n");
    printlist(p);
    printf("\nAfter Multiplication\n");
    p = multiply(head1, head2);
    printlist(p);
}
```



```
PS C:\Users\sheeh\OneDrive\Desktop\C\output> & .\'polynomialaddition.exe'
First polynomial
Enter highest power of polynomial: 2
Enter the coefficient of power 2 10
Enter the coefficient of power 1 20
Enter the coefficient of power 0 30
Second polynomial
Enter highest power of polynomial: 2
Enter the coefficient of power 2 40
Enter the coefficient of power 1 50
Enter the coefficient of power 0 60
10x^2  20x^1  30x^0
40x^2  50x^1  60x^0

After Addition
50x^2  70x^1  90x^0
```



```
PS C:\Users\sheeh\OneDrive\Desktop\C\output> & .\'polynomialaddition.exe'
First polynomial
Enter highest power of polynomial: 2
Enter the coefficient of power 2 10
Enter the coefficient of power 1 20
Enter the coefficient of power 0 30
Enter the coefficient of power 2 10
Enter the coefficient of power 1 20
Enter the coefficient of power 0 30
50x^2  60x^1  70x^0
10x^2  20x^1  30x^0

After Substraction
40x^2  40x^1  40x^0
```



```
PS C:\Users\sheeh\OneDrive\Desktop\C\output> & .\'polynomi
First polynomial
Enter highest power of polynomial: 2
Enter the coefficient of power 2 10
Enter the coefficient of power 1 20
Enter the coefficient of power 0 30
Second polynomial
Enter highest power of polynomial: 2
Enter the coefficient of power 2 40
Enter the coefficient of power 1 50
Enter the coefficient of power 0 60
10x^2  20x^1  30x^0
40x^2  50x^1  60x^0

After Multiplication
400x^4  1300x^3  2800x^2  2700x^1  1800x^0
```