```c
#include<stdio.h>

#include<stdlib.h>

typedef struct node *Nodeptr;

struct node{

    int coeff;

    int exp;

    Nodeptr next;

};

Nodeptr getnode(){

    Nodeptr temp;

    temp = (Nodeptr) malloc(sizeof(struct node));

    return temp;

}
```

```
void attach(int c, int e, Nodeptr *rear){  //Same as InsertLast of union and Intersection
   Nodeptr temp;


   temp = getnode();
   temp->coeff = c;
   temp->exp = e;
   temp->next = NULL;


   (*rear)->next = temp;
   *rear = temp;
}
```

```
Nodeptr CreatePoly(){
   int c,e;


   Nodeptr front,rear,temp;


   rear = getnode();//dummy node
   front = rear;


   printf("Enter the Coefficient [-99 to End]: ");
   scanf("%d",&c);
   while(c!=-99){
      printf("Enter the Exponent: ");
      scanf("%d", &e);
      attach(c,e,&rear);
      printf("Enter the Coefficient [-99 to End]: ");
      scanf("%d",&c);
   }
//free dummy node
   temp=front;
   front = front->next;
   free(temp);


   return front;
}
```

```
void Display(Nodeptr first){

    if (first==NULL)

        return;

     printf("%d",first->coeff);



     if(first->exp>1)

        printf("X%d",first->exp);

    else

        if(first->exp==1)

        printf("X",first->exp);

     if(first->next) printf("+");

     Display(first->next);

}
```

```
int Compare(int x, int y){
    if (x<y)
        return -1;
    else
        if(x==y)
            return 0;
        else
            return 1;
}
```

```
Nodeptr AddPoly(Nodeptr a, Nodeptr b){
  Nodeptr front, rear,temp;
  int sum;

  rear = getnode();  //dummy node
  front = rear;
  while(a && b){
    switch(Compare(a->exp, b->exp)){
      case -1: //a->exp< b-exp
        attach(b->coeff,b->exp, &rear);
        b = b->next;
        break;
      case 0: //a->exp== b-exp
        sum= a->coeff + b->coeff;
        if (sum)
          attach(sum,a->exp, &rear);
        a=a->next;
        b=b->next;
        break;
      case 1://a->exp>b-exp
        attach(a->coeff,a->exp, &rear);
        a=a->next;
        break;
    }
  }
//copy the remaining terms in either a or b

   while(a){

      attach(a->coeff,a->exp, &rear);

      a=a->next;

   }

   while(b){

      attach(b->coeff,b->exp, &rear);

      b=b->next;

   }

//free the dummy node

   temp=front;

   front=front->next;

   free(temp);


   return front;

}

void Erase(Nodeptr *first){

   Nodeptr temp;


   while(*first){

      temp = *first;

      *first = (*first)->next;

      free(temp);

   }

}
```

```
int main(){

    Nodeptr A, B, C;

    A = CreatePoly();
    Display(A);
    printf("\n");

    B = CreatePoly();
    Display(B);
    printf("\n");

    C = AddPoly(A,B);
    Display(C);

    Erase(&A);
    Erase(&B);
    Erase(&C);

}
```