```
struct node
{
	int power;
	int coef;
	struct node *next;
};

void main(void)
{
	struct node *p1,*p2,*p3;
	struct node* create(void);
	struct node* poladd(struct node*, struct node*);
	void print(struct node*);
	clrscr();
	printf("\t\tEntry for the First Polynomial\n");
	p1=create();
	clrscr();
	printf("\t\tEntry for the Second Polynomial\n");
	p2=create();
	clrscr();
	printf("\n\n\n The First Polynomial\n");
	print(p1);
	printf("\n\n\n The Second Polynomial\n");
	print(p2);
	printf("\n\n\n The Addition result\n");
	p3=poladd(p1,p2);
	print(p3);
}
```

```
void print(struct node *prnode)
{
	while(prnode!=NULL)
		{
		printf("%d , %d ;" ,prnode->power, prnode->coef);
		prnode=prnode->next;
		}
}



struct node* create(void)
{
	int i,n;
	struct node *p,*q,*r;
	printf("\nEnter total nodes in this Polynomial");
	scanf("%d",&n);
	if(n>0)
	{
	p=(struct node*)malloc(sizeof(struct node));
	printf("\nEnter the first power and coefficient \n");
	scanf("%d",&(p->power));
	scanf("%d",&(p->coef));
	p->next=NULL;
	}
	q=p;
	for(i=1;i<n;i++)
		{
		r=(struct node*)malloc(sizeof(struct node));
		printf("\nEnter %d power and coefficient\n",i);
		scanf("%d",&(r->power));
		scanf("%d",&(r->coef));
		r->next=NULL;
		q->next=r;
		q=r;
		}
		return(p);
}
```

```
struct node* poladd(struct node *pol1, struct node *pol2)
{
        int newpwr,newcoef;
        struct node *psum,*r,*t;
        t=NULL;
        while((pol1!=NULL)&&(pol2!=NULL))
        {
        if(pol1->power==pol2->power)
                {
                newpwr=pol1->power;
                newcoef=pol1->coef+pol2->coef;
                pol1=pol1->next;
                pol2=pol2->next;
                }
        else if (pol1->power>pol2->power)
                {
                newpwr=pol2->power;
                newcoef=pol2->coef;
                pol2=pol2->next;
                }
            else
            {
                newpwr=pol1->power;
                newcoef=pol1->coef;
                pol1=pol1->next;
            }
        if(newcoef!=0)
        {
          r=(struct node*)malloc(sizeof(struct node));
          r->power=newpwr;
          r->coef=newcoef;
          if(t==NULL)
                psum=r;
          else
                t->next=r;
          t=r;
          r->next=NULL;
        }
  }/*ENDING WHILE LOOP */

/*If either of the lists end.. copy the other one */
if(pol1==NULL)
{
        while(pol2!=NULL)
        {
                r=(struct node*)malloc(sizeof(struct node));
                r->power=pol2->power;
                r->coef=pol2->coef;
                if(t==NULL)
                        psum=r;
                else
                        t->next=r;
                t=r;
                r->next=NULL;
                pol2=pol2->next;
               }
}else
    {
      while(pol1!=NULL)
            {
                r=(struct node*)malloc(sizeof(struct node));
                r->power=pol1->power;
                r->coef=pol1->coef;
                if(t==NULL)
                        psum=r;
                else
                        t->next=r;
                t=r;
                r->next=NULL;
                pol1=pol1->next;
               }
     }
return(psum);
}
```