```
/**********************SUBHAS NATH******************/
/***************************************************/
/***************POLYNOMIAL ADDITION.***************/

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

typedef struct poly
 {
  int coeff;
  int exp;
  poly * next;
 }node;

void main()
 {
  node *head,*head1,*head2;
  void create(node *);
  void show(node *);
  node *add(node *,node *);
  clrscr();
  printf("\n\t\t*****CREATE 1st POLYNOMIAL.*****\n");
  head1=(node *)malloc(sizeof(node));
  create(head1);
  clrscr();
  printf("\n\t\t*****CREATE 2nd POLYNOMIAL.*****\n");
  head2=(node *)malloc(sizeof(node));
  create(head2);
  clrscr();
  printf("\n\t\t*****SHOW 1st POLYNOMIAL.*****\n");
  show(head1);
  printf("\n\t\t*****SHOW 2nd POLYNOMIAL.*****\n");
  show(head2);
  printf("\n\t\t*****AFTER ADDITION.*****\n");
  head=add(head1,head2);
  show(head);
  getch();
 }

void create(node *temp)
 {
  char ch;
  printf("\nEnter the coefficient:-   ");
  scanf("%d",&temp->coeff);
  printf("\nEnter the exponent:-  ");
```

```
 scanf("%d",&temp->exp);
 temp->next=NULL;
 printf("\nWant to continue(y/n):-  ");
 fflush(stdin);
 ch=getchar();
 if(ch!='y')
  return;
 else
  {
   temp->next=(node *)malloc(sizeof(node));
   create(temp->next);
  }
}

void show(node *temp)
 {
 if(temp->next==NULL)
  printf("%dX^%d=0",temp->coeff,temp->exp);
 else
  {
   printf("%dX^%d + ",temp->coeff,temp->exp);
   show(temp->next);
  }
 }

node *add(node *a,node *b)
 {
 node *p,*q,*c,*d,*temp;
 int x;
 void attach(int,int,node *);
 p=a;q=b;
 c=(node *)malloc(sizeof(node));
 d=c;
 while((p!=NULL)&&(q!=NULL))
  {
   if(p->exp==q->exp)
    {
     x=p->coeff+q->coeff;
     if(x!=0)
      {
          attach(x,p->exp,d);
          p=p->next;
          q=q->next;
          temp=d;
          d->next=(node *)malloc(sizeof(node));
          d=d->next;
```

```
      }
     }
    else if(p->exp<q->exp)
     {
      attach(q->coeff,q->exp,d);
      q=q->next;
      temp=d;
      d->next=(node *)malloc(sizeof(node));
      d=d->next;
     }
    else
     {
      attach(p->coeff,p->exp,d);
      p=p->next;
      temp=d;
      d->next=(node *)malloc(sizeof(node));
      d=d->next;
     }
   }
  while(p!=NULL)
   {
    attach(p->coeff,p->exp,d);
    p=p->next;
    temp=d;
    d->next=(node *)malloc(sizeof(node));
    d=d->next;
   }
  while(q!=NULL)
   {
    attach(q->coeff,q->exp,d);
    q=q->next;
    temp=d;
    d->next=(node *)malloc(sizeof(node));
    d=d->next;
   }
  temp->next=NULL;
  d=c;
  return(d);
 }

void attach(int c,int e,node *y)
 {
  y->coeff=c;
  y->exp=e;
 }
```