

# Polynomial Representation

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

struct Term
{
    int coeff;
    int exp;
};
struct Poly
{
    int n;
    struct Term *terms;
};

void create(struct Poly *p)
{
    int i;
    printf("Number of terms?");
    scanf("%d",&p->n);
    p->terms=(struct Term*)malloc(p->n*sizeof(struct
Term));

    printf("Enter terms\n");
    for(i=0;i<p->n;i++)
        scanf("%d%d",&p->terms[i].coeff,&p-
>terms[i].exp);
}

void display(struct Poly p)
{
    int i;
    for(i=0;i<p.n;i++)

printf("%dx%d+",p.terms[i].coeff,p.terms[i].exp);
    printf("\n");
}
```

```

struct Poly *add(struct Poly *p1, struct Poly *p2)
{
    int i, j, k;
    struct Poly *sum;

    sum = (struct Poly *) malloc(sizeof(struct Poly));
    sum->terms = (struct Term *) malloc((p1->n + p2->n) * sizeof(struct Term));
    i = j = k = 0;

    while(i < p1->n && j < p2->n)
    {
        if(p1->terms[i].exp > p2->terms[j].exp)
            sum->terms[k++] = p1->terms[i++];
        else if(p1->terms[i].exp < p2->terms[j].exp)
            sum->terms[k++] = p2->terms[j++];
        else
        {
            sum->terms[k].exp = p1->terms[i].exp;
            sum->terms[k++].coeff = p1->terms[i+
+].coeff + p2->terms[j++].coeff;
        }
    }
    for(; i < p1->n; i++) sum->terms[k++] = p1->terms[i];
    for(; j < p2->n; j++) sum->terms[k++] = p2->terms[j];

    sum->n = k;
    return sum;
}

int main()
{
    struct Poly p1, p2, *p3;

    create(&p1);
    create(&p2);

    p3 = add(&p1, &p2);

```

```
printf("\n");  
display(p1);  
printf("\n");  
display(p2);  
printf("\n");  
display(*p3);
```

```
return 0;
```

```
}
```