```c
struct Polynomial
{
    int coeff;
    int exp;
};

struct Polynomial first[15], second[15], result[15];

void display(struct Polynomial poly[], int terms)
{
    int i;
    printf("\n");
    for(i = 0; i < terms ; i++)
    {
        printf("%dX^%d+ ", poly[i].coeff, poly[i].exp);
    }
}

int readExpression(struct Polynomial poly[])
{
    int terms, i;
    printf("\nNumber of terms: ");
    scanf("%d", &terms);
    printf("\nEnter the coeffecients and exponents in DESCENDING order");
    for(i = 0 ; i<terms; i++)
    {
        printf("\nCoeffecient :");
        scanf("%d", &poly[i].coeff);
        printf("Exponent :");
        scanf("%d", &poly[i].exp);
    }
    return terms;
}

int addExpressions(int firstCount, int secondCount)
{
    int i, j, k;
    i = 0;
    j = 0;
    k = 0;
    while(i < firstCount && j < secondCount)
    {
        if(first[i].exp == second[j].exp)
        {
            result[k].coeff = first[i].coeff + second[j].coeff;
            result[k].exp = first[i].exp;
            i++;
            j++;
            k++;
        }
        else if(first[i].exp > second[j].exp)
        {
            result[k].coeff = first[i].coeff;
            result[k].exp = first[i].exp;
            i++;
            k++;
        }
        else
        {
```

```
            result[k].coeff = second[i].coeff;
            result[k].exp = second[j].exp;
            j++;
            k++;
        }
    }

    while(i < firstCount)
    {
        result[k].coeff = first[i].coeff;
        result[k].exp = first[i].exp;
        k++;
        i++;
    }

    while(j < secondCount)
    {
        result[k].coeff = second[j].coeff;
        result[k].exp = second[j].exp;
        k++;
        j++;
    }
    return k;
}

int main()
{
    int firstCount, secondCount, resultCount;
    printf("\nFirst Expression:\n");
    firstCount = readExpression(first);
    printf("\nSecond Expression:\n");
    secondCount = readExpression(second);
    printf("\nFirst Expression");
    display(first, firstCount);
    display(second, secondCount);
    resultCount = addExpressions(firstCount, secondCount);
    printf("\nResultant Expression:\n");
    display(result, resultCount);
    return 0;
}
```