```
//Created By Ritwik Chandra Pandey on 24/02/21
//183215
//Polynomial operations using Linked List



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

struct polynomial {
   int coeff;
   int exp;
   struct polynomial *next;
};
typedef struct polynomial *poly;
poly head1, head2, result;



poly addTerm(poly head, poly temp) {
   poly p1,p2;
   p1 = p2 = head;

   if (p1 == NULL) {
      head = temp;
   } else {
      while (p1 != NULL && p1 -> exp > temp -> exp) {
         p2 = p1;
         p1 = p1 -> next;
      }
      if (p1 == NULL) {
         p2 -> next = temp;
```

```
        } else if (p1 -> exp == temp -> exp) {
            p1 -> coeff = p1->coeff + temp->coeff;
        } else if (p1 -> exp < temp -> exp) {
            if (p2 == p1) {
                temp -> next = p1;
                head = temp;
            } else {
                temp -> next = p1;
                p2 -> next = temp;
            }
        }
    }
    return head;
}
poly add(poly head1, poly head2) {
    poly t1, t2, sum = NULL, temp = NULL;
    t1 = head1;
    t2 = head2;
    while (t1 != NULL && t2 != NULL) {
        temp = (poly)malloc(sizeof(struct polynomial));
        if (t1 -> exp == t2 -> exp) {
            temp -> coeff = t1 -> coeff + t2 -> coeff;
            temp -> exp = t1 -> exp;
            temp -> next = NULL;
            sum = addTerm(sum,temp);
            t1 = t1 -> next;
            t2 = t2 -> next;
        } else if (t1 -> exp > t2 -> exp) {
            temp -> coeff = t1 -> coeff;
            temp -> exp = t1 -> exp;
            temp -> next = NULL;
            sum = addTerm(sum,temp);
```

```
            t1 = t1 -> next;
        } else {
            temp -> coeff = t2 -> coeff;
            temp -> exp = t2 -> exp;
            temp -> next = NULL;
            sum = addTerm(sum,temp);
            t2 = t2 -> next;
        }
    }
    while (t1 != NULL) {
        temp = (poly)malloc(sizeof(struct polynomial));
        temp -> coeff = t1 -> coeff;
        temp -> exp = t1 -> exp;
        temp -> next = NULL;
        sum = addTerm(sum,temp);
        t1 = t1 -> next;
    }
    while (t2 != NULL) {
        temp = (poly)malloc(sizeof(struct polynomial));
        temp -> coeff = t2 -> coeff;
        temp -> exp = t2 -> exp;
        temp -> next = NULL;
        sum = addTerm(sum,temp);
        t2 = t2 -> next;
    }
    return sum;
}
poly create(poly head) {
    poly temp;
    char ch='y';
    int coeff, exp;
```

```
    do{
        temp = (poly)malloc(sizeof(struct polynomial));
        printf("Enter coeff: ");
        scanf("%d", &coeff);
        temp -> coeff = coeff;
        printf("Enter exp: ");
        scanf("%d", &exp);
        temp -> exp = exp;
        temp -> next = NULL;
        head = addTerm(head, temp);
    printf("Want to add further?(y|n) : ");
    scanf(" %c",&ch);

}while(ch!='n');

    return head;
}
poly mul(poly head1,poly head2) {
    poly t1, t2, temp, pro;
    t1 = t2 = temp = pro = NULL;
    for(t1 = head1; t1 != NULL; t1 = t1 -> next) {
        for(t2 = head2; t2 != NULL; t2 = t2 -> next) {
            temp = (poly)malloc(sizeof(struct polynomial));
            temp -> coeff = t1 -> coeff * t2 -> coeff;
            temp -> exp = t1 -> exp + t2 -> exp;
            temp -> next = NULL;
            pro = addTerm(pro,temp);
        }
    }
    return pro;
}
```

```
poly sub(poly head1, poly head2) {
    poly t1, t2, sub = NULL, temp = NULL;
    t1 = head1;
    t2 = head2;
    while (t1 != NULL && t2 != NULL) {
        temp = (poly)malloc(sizeof(struct polynomial));
        if (t1 -> exp == t2 -> exp) {
            temp -> coeff = t1 -> coeff - t2 -> coeff;
            temp -> exp = t1 -> exp;
            temp -> next = NULL;
            sub = addTerm(sub,temp);
            t1 = t1 -> next;
            t2 = t2 -> next;
        } else if (t1 -> exp > t2 -> exp) {
            temp -> coeff = t1 -> coeff;
            temp -> exp = t1 -> exp;
            temp -> next = NULL;
            sub = addTerm(sub,temp);
            t1 = t1 -> next;
        } else {
            temp -> coeff = -1 * t2 -> coeff;
            temp -> exp = t2 -> exp;
            temp -> next = NULL;
            sub = addTerm(sub,temp);
            t2 = t2 -> next;
        }
    }
    while (t1 != NULL) {
        temp = (poly)malloc(sizeof(struct polynomial));
        temp -> coeff = t1 -> coeff;
        temp -> exp = t1 -> exp;
        temp -> next = NULL;
```

```
        sub = addTerm(sub,temp);
        t1 = t1 -> next;
    }
    while (t2 != NULL) {
        temp = (poly)malloc(sizeof(struct polynomial));
        temp -> coeff = -1 * t2 -> coeff;
        temp -> exp = t2 -> exp;
        temp -> next = NULL;
        sub = addTerm(sub,temp);
        t2 = t2 -> next;
    }
    return sub;
}
void print(poly head) {
    poly temp = head;
    while (temp != NULL) {
        printf("%d X^ %d --->", temp -> coeff, temp -> exp);
        temp = temp -> next;
    }
    printf("NULL\n");
}


int main(){
    int selection;

    head1 = head2 = result = NULL;
    printf("\n\tPolynomial Operations Using Linked List\n\t");
    printf("Enter 1st polynomial: \n");
    head1 = NULL;
    head1 = create(head1);
    printf("Enter 2nd polynomial: \n");
```

```
head2 = NULL;
head2 = create(head2);

do{
    printf("\n\t1.ADDITION\n\t2.SUBTRACTION\n\t3.MULTIPLICATION\n\t4.DISPLAY\n\t5.EXIT\n");
    printf("Enter your choice\n");
    scanf("%d",&selection);


    switch(selection){



        case 1:
            result = NULL;
            result = add(head1,head2);
            print(result);
            break;
        case 2:
            result =NULL;
            result = sub(head1,head2);
            print(result);
            break;


        case 3:
            result=NULL;
            result= mul(head1,head2);
            print(result);
            break;
        case 4:
            print(head1);
```

```
            print(head2);
            break;
        case 5:
            exit(1);
        default:
            printf("Please enter correct choice!\n");
    }}while(selection!=5);
}
```