**DSA-LAB ASSIGNMENT 2**

**(Name:Sneha Kondawar)**

**(Roll No:BT22CSE075)**

**Question 1:**

CODE:

#include<stdio.h>

#include<stdlib.h>

struct node{

    int coeff;

    int expo;

    struct node \* next;

};

struct node \*insert(struct node \*head, int co, int ex) {

    struct node \*temp;

    struct node \* new = malloc(sizeof(struct node));

    new->coeff = co;

    new->expo = ex;

    new->next = NULL;

    if (head == NULL || ex > head->expo) {

        new->next = head;

        head = new;

    }

    else if(ex == head->expo){

        head->coeff +=co;

        free(new);

    }

    else {

        temp = head;

        while (temp->next != NULL && ex < temp->next->expo) {

            temp = temp->next;

        }

        if (temp->next != NULL && ex == temp->next->expo) {

            temp->next->coeff += co;

            free(new);

        }

        else{

        new->next = temp->next;

        temp->next = new;

    }

    }

    return head;

}

struct node \*create(struct node \*head, int n) {

    int coeff;

    int expo;

    for (int i = 0; i < n; i++) {

        printf("Enter the coefficient of term %d: ", i + 1);

        scanf("%d", &coeff);

        printf("Enter the exponent of term %d: ", i + 1);

        scanf("%d", &expo);

        head = insert(head, coeff, expo);

    }

    return head;

}

void print(struct node \*head) {

    struct node \*temp = head;

    if (head == NULL) {

        printf("No polynomial.");

        return;

    }

    while (temp != NULL) {

        printf("%dx^%d ", temp->coeff, temp->expo);

        if (temp->next != NULL) {

            printf(" + ");

        }

        temp = temp->next;

    }

}

struct node \* multiply(struct node \* head1,struct node \*head2){

    struct node \* head=NULL;

    struct node \* ptr1=head1;

    struct node \* ptr2=head2;

    int res1,res2;

    while(ptr1!=NULL){

        while(ptr2!=NULL){

            res1=ptr1->coeff\*ptr2->coeff;

            res2=ptr1->expo+ptr2->expo;

            head=insert(head,res1,res2);

            ptr2=ptr2->next;

        }

        ptr1=ptr1->next;

        ptr2=head2;

    }

    return head;

}

int main(){

    struct node \*head1 = NULL;

    struct node \*head2 = NULL;

    struct node \*head3 = NULL;

    int n, m;

    printf("Enter the number of terms in polynomial 1: ");

    scanf("%d", &n);

    head1 = create(head1, n);

    printf("Enter the number of terms in polynomial 2: ");

    scanf("%d", &m);

    head2 = create(head2, m);

    print(head1);

    printf(" \n");

    print(head2);

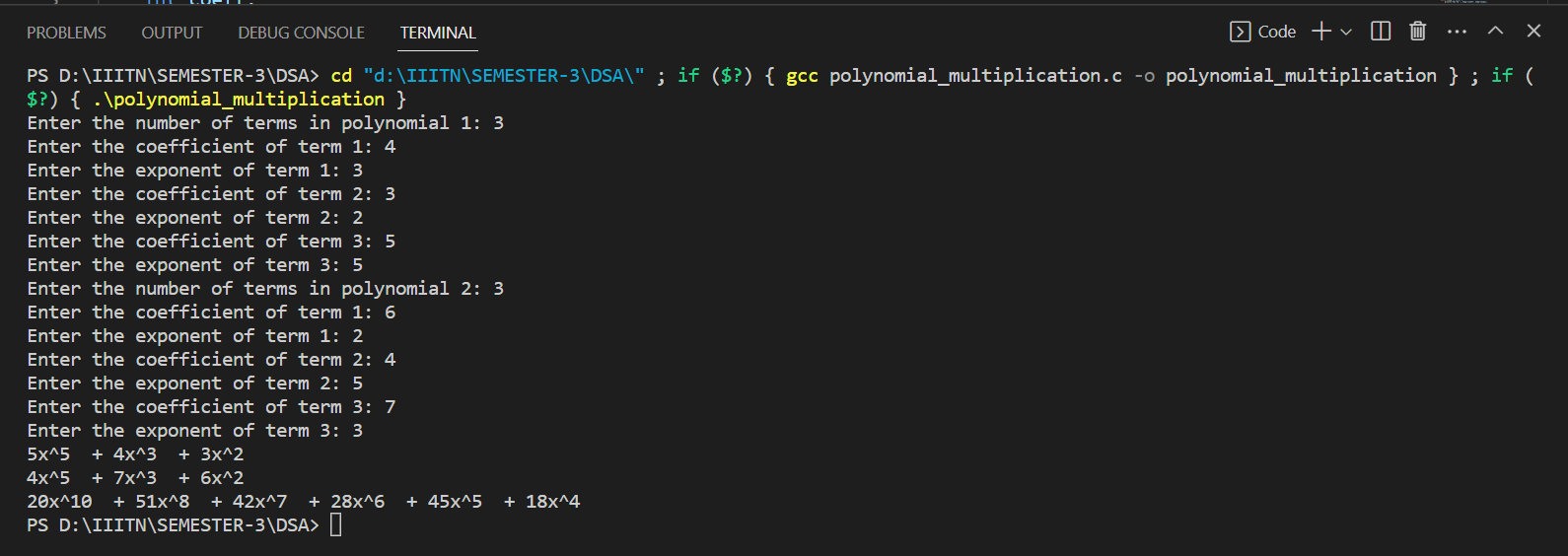
    printf(" \n");

    head3=multiply(head1,head2);

    print(head3);

    return 0;

}

OUTPUT:

**Question 2:**

**CODE:**

#include<stdio.h>

#include<stdlib.h>

struct node

{

    long int digit;

    struct node \*next;

};

struct node \* push(struct node \* head,int val){

    struct node \* new=malloc(sizeof(struct node));

    new->digit=val;

    new->next=head;

    head=new;

    return head;

}

struct node \* add(struct node \* head1,struct node \* head2){

    if(head1->digit==0){

        return head1;

    }

    if(head2->digit==0){

        return head2;

    }

    struct node \* ptr1=head1;

    struct node \*ptr2=head2;

    struct node \* head3=NULL;

    int sum=0,carry=0;

    while(ptr1||ptr2){

        sum=0;

        if(ptr1!=NULL){

            sum=sum+ptr1->digit;

        }

        if(ptr2!=NULL){

            sum=sum+ptr2->digit;

        }

        sum=sum+carry;

        carry=sum/10;

        sum=sum%10;

        head3=push(head3,sum);

        if(ptr1!=NULL){

            ptr1=ptr1->next;

        }

        if(ptr2!=NULL){

            ptr2=ptr2->next;

        }

    }

    if(carry){

        head3=push(head3,carry);

    }

    return head3;

}

struct node \* reversell(struct node \* head){

    if(head==NULL){

        return head;

    }

    struct node \* current=NULL;

    struct node \* next=head->next;

    head->next=NULL;

    while(next!=NULL){

        current=next;

        next=next->next;

        current->next=head;

        head=current;

    }

    return head;

}

struct node \* add\_node(struct node \* head,int val){

    struct node \*new=malloc(sizeof(struct node));

    new->digit=val;

    new->next=NULL;

    new->next=head;

    head=new;

    return head;

}

struct node \* create(struct node \*head,int n){

    while (n!=0){

        head=add\_node(head,n%10);

        n=n/10;

    }

    return head;

}

void print\_nodes(struct node \* head){

    struct node\* temp=head;

        while(temp!=NULL){

            printf("%d",temp->digit);

            temp=temp->next;

        }

}

int main()

{

    struct node \*head1,\*head2;

    head1 = NULL;

    head2=NULL;

    long int n1,n2;

    printf("\nEnter the number 1: ");

    scanf("%d",&n1);

    printf("\nEnter the number 2: ");

    scanf("%d",&n2);

    head1=create(head1,n1);

    print\_nodes(head1);

    printf("\n");

    head2=create(head2,n2);

    print\_nodes(head2);

    head1=reversell(head1);

    head2=reversell(head2);

    printf("\nReversed Linked lists: \n");

    print\_nodes(head1);

    printf("\n");

    print\_nodes(head2);

    struct node \* head3=NULL;

    head3=add(head1,head2);

    printf("\n");

    printf("\n");

    printf("Sum of the two long integers is: ");

    print\_nodes(head3);

    return 0;

}

**OUTPUT:**

****