**DSA Lab Questions**

**LAB 1**

Q 1

**Code**

#include <stdio.h>

#include <stdlib.h>

struct Term {

    int coefficient;

    int exponent;

    struct Term\* next;

};

struct Term\* createTerm(int coefficient, int exponent) {

    struct Term\* newTerm = (struct Term\*)malloc(sizeof(struct Term));

    newTerm->coefficient = coefficient;

    newTerm->exponent = exponent;

    newTerm->next = NULL;

    return newTerm;

}

void insertTerm(struct Term\*\* poly, int coefficient, int exponent) {

    struct Term\* newTerm = createTerm(coefficient, exponent);

    if (\*poly == NULL || (\*poly)->exponent < exponent) {

        newTerm->next = \*poly;

        \*poly = newTerm;

    } else {

        struct Term\* temp = \*poly;

        while (temp->next != NULL && temp->next->exponent > exponent) {

            temp = temp->next;

        }

        newTerm->next = temp->next;

        temp->next = newTerm;

    }

}

struct Term\* addPolynomials(struct Term\* poly1, struct Term\* poly2) {

    struct Term\* result = NULL;

    struct Term\* temp1 = poly1;

    struct Term\* temp2 = poly2;

    while (temp1 != NULL && temp2 != NULL) {

        if (temp1->exponent == temp2->exponent) {

            insertTerm(&result, temp1->coefficient + temp2->coefficient, temp1->exponent);

            temp1 = temp1->next;

            temp2 = temp2->next;

        } else if (temp1->exponent > temp2->exponent) {

            insertTerm(&result, temp1->coefficient, temp1->exponent);

            temp1 = temp1->next;

        } else {

            insertTerm(&result, temp2->coefficient, temp2->exponent);

            temp2 = temp2->next;

        }

    }

    while (temp1 != NULL) {

        insertTerm(&result, temp1->coefficient, temp1->exponent);

        temp1 = temp1->next;

    }

    while (temp2 != NULL) {

        insertTerm(&result, temp2->coefficient, temp2->exponent);

        temp2 = temp2->next;

    }

    return result;

}

void displayPolynomial(struct Term\* poly) {

    struct Term\* temp = poly;

    while (temp != NULL) {

        printf("%dx^%d", temp->coefficient, temp->exponent);

        temp = temp->next;

        if (temp != NULL) {

            printf(" + ");

        }

    }

    printf("\n");

}

int main() {

    int n1, n2, coefficient, exponent;

    struct Term\* poly1 = NULL;

    struct Term\* poly2 = NULL;

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

    scanf("%d", &n1);

    printf("Enter the terms (coefficient exponent) for polynomial A:\n");

    for (int i = 0; i < n1; i++) {

        scanf("%d %d", &coefficient, &exponent);

        insertTerm(&poly1, coefficient, exponent);

    }

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

    scanf("%d", &n2);

    printf("Enter the terms (coefficient exponent) for polynomial B:\n");

    for (int i = 0; i < n2; i++) {

        scanf("%d %d", &coefficient, &exponent);

        insertTerm(&poly2, coefficient, exponent);

    }

    struct Term\* result = addPolynomials(poly1, poly2);

    printf("Polynomial A: ");

    displayPolynomial(poly1);

    printf("Polynomial B: ");

    displayPolynomial(poly2);

    printf("Resultant Polynomial: ");

    displayPolynomial(result);

    while (poly1 != NULL) {

        struct Term\* temp = poly1;

        poly1 = poly1->next;

        free(temp);

    }

    while (poly2 != NULL) {

        struct Term\* temp = poly2;

        poly2 = poly2->next;

        free(temp);

    }

    while (result != NULL) {

        struct Term\* temp = result;

        result = result->next;

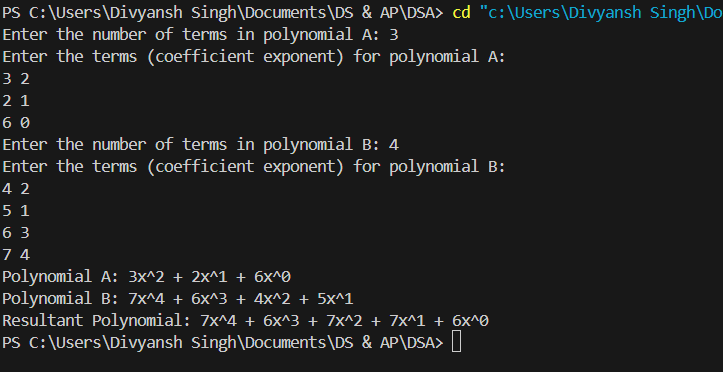
        free(temp);

    }

    return 0;

}

**OUTPUT**

****

Q2

**Code**

#include <stdio.h>

#include <stdlib.h>

struct Term {

    int coefficient;

    int exponent;

    struct Term\* next;

};

struct Term\* createTerm(int coefficient, int exponent) {

    struct Term\* newTerm = (struct Term\*)malloc(sizeof(struct Term));

    newTerm->coefficient = coefficient;

    newTerm->exponent = exponent;

    newTerm->next = NULL;

    return newTerm;

}

void insertTerm(struct Term\*\* poly, int coefficient, int exponent) {

    if (coefficient == 0) {

        return;

    }

    struct Term\* newTerm = createTerm(coefficient, exponent);

    if (\*poly == NULL || (\*poly)->exponent < exponent) {

        newTerm->next = \*poly;

        \*poly = newTerm;

    } else if ((\*poly)->exponent == exponent) {

        (\*poly)->coefficient += coefficient;

        if ((\*poly)->coefficient == 0) {

            struct Term\* temp = \*poly;

            \*poly = (\*poly)->next;

            free(temp);

        }

        free(newTerm);

    } else {

        struct Term\* temp = \*poly;

        while (temp->next != NULL && temp->next->exponent > exponent) {

            temp = temp->next;

        }

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

            temp->next->coefficient += coefficient;

            if (temp->next->coefficient == 0) {

                struct Term\* toDelete = temp->next;

                temp->next = temp->next->next;

                free(toDelete);

            }

            free(newTerm);

        } else {

            newTerm->next = temp->next;

            temp->next = newTerm;

        }

    }

}

void displayPolynomial(struct Term\* poly) {

    struct Term\* temp = poly;

    while (temp != NULL) {

        printf("%dx^%d", temp->coefficient, temp->exponent);

        temp = temp->next;

        if (temp != NULL) {

            printf(" + ");

        }

    }

    printf("\n");

}

int main() {

    int n1, n2, coefficient, exponent;

    struct Term\* poly = NULL;

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

    scanf("%d", &n1);

    printf("Enter the terms (coefficient exponent) for polynomial A:\n");

    for (int i = 0; i < n1; i++) {

        scanf("%d %d", &coefficient, &exponent);

        insertTerm(&poly, coefficient, exponent);

    }

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

    scanf("%d", &n2);

    printf("Enter the terms (coefficient exponent) for polynomial B:\n");

    for (int i = 0; i < n2; i++) {

        scanf("%d %d", &coefficient, &exponent);

        insertTerm(&poly, coefficient, exponent);

    }

    printf("Resultant Polynomial: ");

    displayPolynomial(poly);

    while (poly != NULL) {

        struct Term\* temp = poly;

        poly = poly->next;

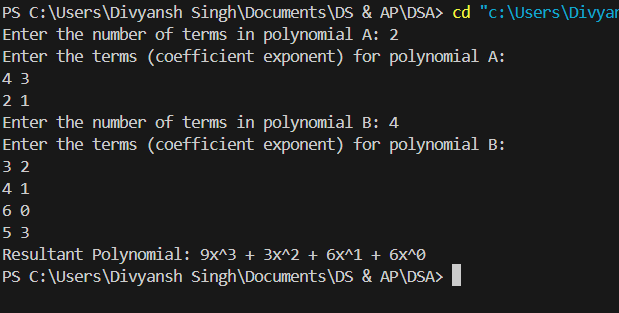
        free(temp);

    }

    return 0;

}

**OUTPUT**

****