DSA LAB

Assignment 6

Name: Madhav Jha
Roll no.: E3-48
Branch: CSE (AI & ML)

Polynomial addition using LinkedList

Statement:

Implement a program to add two polynomials represented as linked lists A and B to get resultant polynomial represented as linked lists C. Make use of function add_Polynomial() to add A and B. Starting addresses of A and B should not be declared global and should be passed to function add_Polynomial(). add_Polynomial() will return the resultant polynomial C.

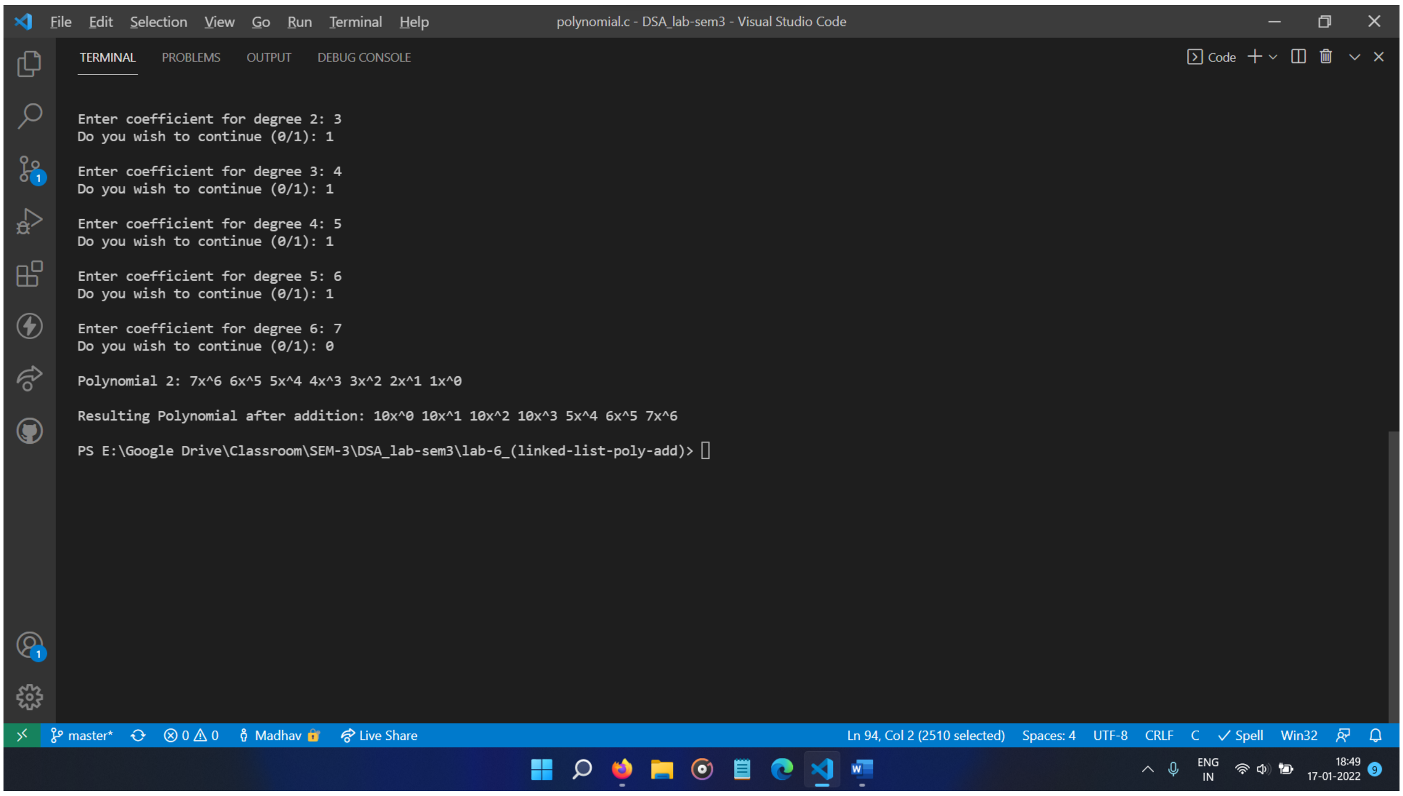
Code:

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

struct LinkedNode {
    int coeff;
    int power;
    struct LinkedNode* next;
};

void addNode(struct LinkedNode** head, int coeff, int power) {
    struct LinkedNode* node = NULL;
    node = (struct LinkedNode*)malloc(sizeof(struct LinkedNode));
    node->coeff = coeff;
    node->power = power;
    node->next = *head;
    *head = node;
}

void printPolynomial(struct LinkedNode** head) {

    struct LinkedNode* temp = *head;
    while (temp != NULL) {
        printf("%dx^%d ", temp->coeff, temp->power);
        temp = temp->next;
    }
    printf("\n");
}

void takeInput(struct LinkedNode** poly) {
```

```c
    int inp = 1, count = 0;
    do {
        int coeff = 0;
        printf("\nEnter coefficient for degree %d: ", count);
        scanf("%d", &coeff);

        addNode(poly, coeff, count);
        ++count;

        printf("Do you wish to continue (0/1): ");
        scanf("%d", &inp);
    } while (inp != 0);
}

struct LinkedNode** addPoly(struct LinkedNode** poly1, struct
LinkedNode** poly2) {
    struct LinkedNode** res = (struct
LinkedNode**)malloc(sizeof(struct LinkedNode*));
    *res = NULL;

    struct LinkedNode* p1 = *poly1;
    struct LinkedNode* p2 = *poly2;

    while (p1 != NULL && p2 != NULL) {
        if (p1->power == p2->power) {
            addNode(res, p1->coeff + p2->coeff, p1->power);
            p1 = p1->next;
            p2 = p2->next;
        }
        else if (p1->power > p2->power) {
            addNode(res, p1->coeff, p1->power);
            p1 = p1->next;
        }
        else {
            addNode(res, p2->coeff, p2->power);
            p2 = p2->next;
        }
    }
    return res;
}
```

```c
int main() {

    struct LinkedNode** poly1 = (struct
LinkedNode**)malloc(sizeof(struct LinkedNode*));
    struct LinkedNode** poly2 = (struct
LinkedNode**)malloc(sizeof(struct LinkedNode*));
    struct LinkedNode** poly3 = (struct
LinkedNode**)malloc(sizeof(struct LinkedNode*));
    *poly1 = NULL;
    *poly2 = NULL;
    *poly3 = NULL;

    printf("\n\nEnter for polynomial 1:");
    takeInput(poly1);
    printf("\nPolynomial 1: ");
    printPolynomial(poly1);

    printf("\n\nEnter for polynomial 2:");
    takeInput(poly2);
    printf("\nPolynomial 2: ");
    printPolynomial(poly2);

    poly3 = addPoly(poly1, poly2);
    printf("\nResulting Polynomial after addition: ");
    printPolynomial(poly3);
    printf("\n");

    return 0;
}
```

Output:



File Edit Selection View Go Run Terminal Help
polynomial.c - DSA_lab-sem3 - Visual Studio Code
TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE
Code
master*
⊗ 0 ⚠ 0
Madhav
Live Share
Ln 94, Col 2 (2510 selected) Spaces: 4 UTF-8 CRLF C ✓ Spell Win32
ENG IN
18:49 17-01-2022

```
Enter coefficient for degree 2: 3
Do you wish to continue (0/1): 1

Enter coefficient for degree 3: 4
Do you wish to continue (0/1): 1

Enter coefficient for degree 4: 5
Do you wish to continue (0/1): 1

Enter coefficient for degree 5: 6
Do you wish to continue (0/1): 1

Enter coefficient for degree 6: 7
Do you wish to continue (0/1): 0

Polynomial 2: 7x^6 6x^5 5x^4 4x^3 3x^2 2x^1 1x^0

Resulting Polynomial after addition: 10x^0 10x^1 10x^2 10x^3 5x^4 6x^5 7x^6

PS E:\Google Drive\Classroom\SEM-3\DSA_lab-sem3\lab-6_(linked-list-poly-add)> 
```