**Deletion in a Linked List**

#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node\* next;

};

struct Node\* createNode(int value) {

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));

if (newNode == NULL) {

printf("Memory allocation failed.\n");

exit(1);

}

newNode->data = value;

newNode->next = NULL;

return newNode;

}

void insertNode(struct Node\*\* head, int value) {

struct Node\* newNode = createNode(value);

if (\*head == NULL) {

\*head = newNode;

return;

}

struct Node\* current = \*head;

while (current->next != NULL) {

current = current->next;

}

current->next = newNode;

}

void deleteNode(struct Node\*\* head, int value) {

if (\*head == NULL) {

printf("Linked list is empty.\n");

return;

}

struct Node\* current = \*head;

struct Node\* prev = NULL;

while (current != NULL && current->data != value) {

prev = current;

current = current->next;

}

if (current == NULL) {

printf("Node with value %d not found.\n", value);

return;

}

if (prev == NULL) {

\*head = current->next;

} else {

prev->next = current->next;

}

free(current);

printf("Node with value %d deleted.\n", value);

}

void printList(struct Node\* head) {

struct Node\* current = head;

while (current != NULL) {

printf("%d -> ", current->data);

current = current->next;

}

printf("NULL\n");

}

int main() {

struct Node\* head = NULL;

int n, value;

printf("Enter the number of nodes: ");

scanf("%d", &n);

for (int i = 0; i < n; i++) {

printf("Enter value for node %d: ", i + 1);

scanf("%d", &value);

insertNode(&head, value);

}

printf("Linked List: ");

printList(head);

printf("Enter the value of the node to delete: ");

scanf("%d", &value);

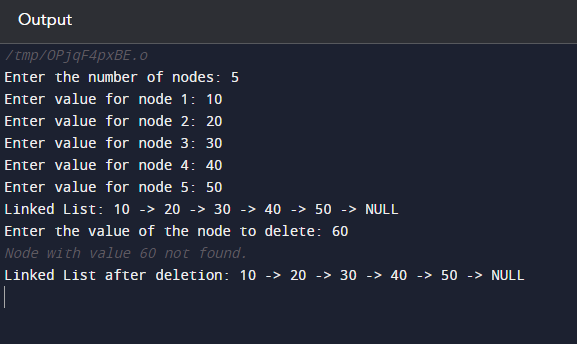
deleteNode(&head, value);

printf("Linked List after deletion: ");

printList(head);

return 0;

}



**Accept a polynomial**

#include <stdio.h>

#include <stdlib.h>

struct Term {

int coefficient;

int exponent;

struct Term\* next;

};

struct Term\* createTerm(int coeff, int exp) {

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

if (newTerm == NULL) {

printf("Memory allocation failed.\n");

exit(1);

}

newTerm->coefficient = coeff;

newTerm->exponent = exp;

newTerm->next = NULL;

return newTerm;

}

void insertTerm(struct Term\*\* head, int coeff, int exp) {

struct Term\* newTerm = createTerm(coeff, exp);

if (\*head == NULL) {

\*head = newTerm;

return;

}

struct Term\* current = \*head;

while (current->next != NULL) {

current = current->next;

}

current->next = newTerm;

}

void printPolynomial(struct Term\* head) {

struct Term\* current = head;

int isFirstTerm = 1;

while (current != NULL) {

if (current->coefficient != 0) {

if (!isFirstTerm) {

if (current->coefficient > 0) {

printf(" + ");

} else {

printf(" - ");

}

}

int absCoeff = abs(current->coefficient);

if (current->exponent == 0) {

printf("%d", absCoeff);

} else {

if (absCoeff != 1) {

printf("%dx^%d", absCoeff, current->exponent);

} else {

printf("x^%d", current->exponent);

}

}

isFirstTerm = 0;

}

current = current->next;

}

printf("\n");

}

int main() {

struct Term\* polynomial = NULL;

int n, coeff, exp;

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

scanf("%d", &n);

for (int i = 0; i < n; i++) {

printf("Enter coefficient and exponent for term %d: ", i + 1);

scanf("%d %d", &coeff, &exp);

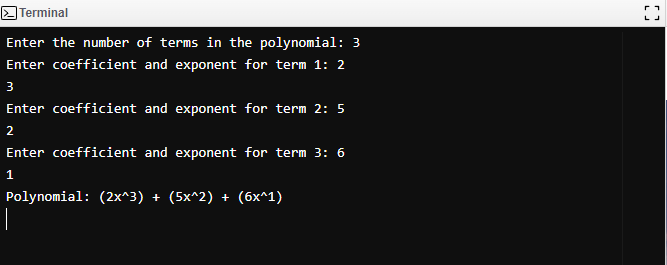
insertTerm(&polynomial, coeff, exp);

}

printf("Polynomial: ");

printPolynomial(polynomial);

return 0;

}

**Addition or subtraction of two polynomials**

#include <stdio.h>

#include <stdlib.h>

struct Term {

int coefficient;

int exponent;

struct Term\* next;

};

struct Term\* createTerm(int coeff, int exp) {

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

if (newTerm == NULL) {

printf("Memory allocation failed.\n");

exit(1);

}

newTerm->coefficient = coeff;

newTerm->exponent = exp;

newTerm->next = NULL;

return newTerm;

}

void insertTerm(struct Term\*\* head, int coeff, int exp) {

struct Term\* newTerm = createTerm(coeff, exp);

if (\*head == NULL) {

\*head = newTerm;

return;

}

struct Term\* current = \*head;

while (current->next != NULL) {

current = current->next;

}

current->next = newTerm;

}

void printPolynomial(struct Term\* head) {

struct Term\* current = head;

int isFirstTerm = 1;

while (current != NULL) {

if (current->coefficient != 0) {

if (!isFirstTerm) {

if (current->coefficient > 0) {

printf(" + ");

} else {

printf(" - ");

}

}

int absCoeff = abs(current->coefficient);

if (current->exponent == 0) {

printf("%d", absCoeff);

} else {

if (absCoeff != 1) {

printf("%dx^%d", absCoeff, current->exponent);

} else {

printf("x^%d", current->exponent);

}

}

isFirstTerm = 0;

}

current = current->next;

}

}

void performOperation(struct Term\* poly1, struct Term\* poly2, char operation) {

struct Term\* result = NULL;

struct Term\* current1 = poly1;

struct Term\* current2 = poly2;

while (current1 != NULL || current2 != NULL) {

int coeff1 = (current1 != NULL) ? current1->coefficient : 0;

int exp1 = (current1 != NULL) ? current1->exponent : 0;

int coeff2 = (current2 != NULL) ? current2->coefficient : 0;

int exp2 = (current2 != NULL) ? current2->exponent : 0;

if (exp1 > exp2) {

insertTerm(&result, coeff1, exp1);

current1 = current1->next;

} else if (exp1 < exp2) {

if (operation == '-') {

coeff2 = -coeff2;

}

insertTerm(&result, coeff2, exp2);

current2 = current2->next;

} else {

int newCoeff = (operation == '+') ? (coeff1 + coeff2) : (coeff1 - coeff2);

insertTerm(&result, newCoeff, exp1);

current1 = current1->next;

current2 = current2->next;

}

}

printf("Result: ");

printPolynomial(result);

printf("\n");

}

int main() {

struct Term\* polynomial1 = NULL;

struct Term\* polynomial2 = NULL;

int n1, n2, coeff, exp;

char operation;

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

scanf("%d", &n1);

for (int i = 0; i < n1; i++) {

printf("Enter coefficient and exponent for term %d of first polynomial: ", i + 1);

scanf("%d %d", &coeff, &exp);

insertTerm(&polynomial1, coeff, exp);

}

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

scanf("%d", &n2);

for (int i = 0; i < n2; i++) {

printf("Enter coefficient and exponent for term %d of second polynomial: ", i + 1);

scanf("%d %d", &coeff, &exp);

insertTerm(&polynomial2, coeff, exp);

}

printf("Enter the operation (+ for addition, - for subtraction): ");

scanf(" %c", &operation);

printf("First Polynomial: ");

printPolynomial(polynomial1);

printf("\n");

printf("Second Polynomial: ");

printPolynomial(polynomial2);

printf("\n");

printf("Operation: ");

performOperation(polynomial1, polynomial2, operation);

return 0;

}

A screenshot of a computer

Description automatically generated