

Basic Linked List - basic operations and some application

Creation, Insertion, Deletion, Traversal, Searching an element.

Implementing types of LL- doubly LL, circular LL, circular doubly LL etc

1. polynomial operations using LL

```
2. #include<stdio.h>
3. #include<stdlib.h>
4. struct Node
5. {
6.     int coeff;
7.     int pow;
8.     struct Node* next;
9. };
10. void readPolynomial(struct Node** poly)
11. {
12.     int coeff, exp, cont;
13.     struct Node* temp = (struct Node*)malloc(sizeof(struct Node));
14.     *poly = temp;
15.     do{
16.         printf("\n Coeffecient: ");
17.         scanf("%d", &coeff);
18.         printf("\n Exponent: ");
19.         scanf("%d", &exp);
20.         temp->coeff = coeff;
21.         temp->pow = exp;
22.         temp->next = NULL;
23.         printf("\nHave more terms? 1 for y and 0 for no: ");
24.         scanf("%d", &cont);
25.         if(cont)
26.         {
27.             temp->next = (struct Node*)malloc(sizeof(struct Node));
28.             temp = temp->next;
29.             temp->next = NULL;
30.         }
31.     }while(cont);
32. }
33. void displayPolynomial(struct Node* poly)
34. {
35.     printf("\nPolynomial expression is: ");
36.     while(poly != NULL)
37.     {
38.         printf("%dX^%d", poly->coeff, poly->pow);
39.         poly = poly->next;
```

```

40.         if(poly != NULL)
41.             printf("+");
42.     }
43.}
44.void addPolynomials(struct Node** result, struct Node* first, struct
    Node* second)
45.{
46.    struct Node* temp = (struct Node*)malloc(sizeof(struct Node));
47.    temp->next = NULL;
48.    *result = temp;
49.    while(first && second)
50.    {
51.        if(first->pow > second->pow)
52.        {
53.            temp->coeff = first->coeff;
54.            temp->pow = first->pow;
55.            first = first->next;
56.        }
57.        else if(first->pow < second->pow)
58.        {
59.            temp->coeff = second->coeff;
60.            temp->pow = second->pow;
61.            second = second->next;
62.        }
63.        else
64.        {
65.            temp->coeff = first->coeff + second->coeff;
66.            temp->pow = first->pow;
67.            first = first->next;
68.            second = second->next;
69.        }
70.        if(first && second)
71.        {
72.            temp->next = (struct Node*)malloc(sizeof(struct Node));
73.            temp = temp->next;
74.            temp->next = NULL;
75.        }
76.    }
77.    while(first || second)
78.    {
79.        temp->next = (struct Node*)malloc(sizeof(struct Node));
80.        temp = temp->next;
81.        temp->next = NULL;
82.        if(first)
83.        {
84.            temp->coeff = first->coeff;
85.            temp->pow = first->pow;
86.            first = first->next;

```

```

87.     }
88.     else if(second)
89.     {
90.         temp->coeff = second->coeff;
91.         temp->pow = second->pow;
92.         second = second->next;
93.     }
94. }
95.}
96.int main()
97.{
98.    struct Node* first = NULL;
99.    struct Node* second = NULL;
100.    struct Node* result = NULL;
101.    printf("\nEnter the corresponding data:-\n");
102.    printf("\nFirst polynomial:\n");
103.    readPolynomial(&first);
104.    displayPolynomial(first);
105.    printf("\nSecond polynomial:\n");
106.    readPolynomial(&second);
107.    displayPolynomial(second);
108.    addPolynomials(&result, first, second);
109.    displayPolynomial(result);
110.    return 0;
111.}

```

Output

```

/tmp/HLNGjYHEJN.o
Enter the corresponding data:-

First polynomial:

Coeffecient: 5
Exponent: 3
Have more terms? 1 for y and 0 for no: 1
Coeffecient: 6
Exponent: 7
Have more terms? 1 for y and 0 for no: 0
Polynomial expression is: 5X^3+6X^7
Second polynomial:

Coeffecient: 2
Exponent: 3
Have more terms? 1 for y and 0 for no: 1
Coeffecient: 4
Exponent: 3
Have more terms? 1 for y and 0 for no: 0
Polynomial expression is: 2X^3+4X^3
Polynomial expression is: 7X^3+6X^7+4X^3

```

2. admission scenario with admissions and cancellations

```
#include <stdio.h>
void main()
{   int p,c,m,t,mp;
    printf("Eligibility Criteria for Admission in College :\n");
    printf("Marks in Maths >=65\n");
    printf("and Marks in Phy >=55\n");
    printf("and Marks in Chem>=50\n");
    printf("and Total in all three subject >=170\n");
    printf("or Total in Maths and Physics >=120\n");
    printf("-----\n");

    printf("Input the marks obtained in Physics :");
    scanf("%d",&p);
    printf("Input the marks obtained in Chemistry :");
    scanf("%d",&c);
    printf("Input the marks obtained in Mathematics :");
    scanf("%d",&m);
    printf("Total marks of Maths, Physics and Chemistry : %d\n",m+p+c);
    printf("Total marks of Maths and Physics : %d\n",m+p);
    if (m>=65)
        if(p>=55)
            if(c>=50)
                if((m+p+c)>=170||(m+p)>=120)
                    printf("The candidate is eligible for admission.\n");
                else
                    printf("The candidate is not eligible for admission.\n");
            else
                printf("The candidate is not eligible for admissio.\n");
        else
            printf("The candidate is not eligible for admission.\n");
    else
        printf("The candidate is not eligible for admission.\n");
}
```

Output

```
/tmp/a4C0xSE2ad.o
Eligibility Criteria for Admission in College :
Marks in Maths >=65
and Marks in Phy >=55
and Marks in Chem>=50
and Total in all three subject >=170
or Total in Maths and Physics >=120
-----
Input the marks obtained in Physics :80
Input the marks obtained in Chemistry :75
Input the marks obtained in Mathematics :96
Total marks of Maths, Physics and Chemistry : 251
Total marks of Maths and Physics : 176
The candidate is eligible for admission.
```

Output

```
/tmp/a4C0xSE2ad.o
Eligibility Criteria for Admission in College :
Marks in Maths >=65
and Marks in Phy >=55
and Marks in Chem>=50
and Total in all three subject >=170
or Total in Maths and Physics >=120
-----
Input the marks obtained in Physics :35
Input the marks obtained in Chemistry :45
Input the marks obtained in Mathematics :50
Total marks of Maths, Physics and Chemistry : 130
Total marks of Maths and Physics : 85
The candidate is not eligible for admission.
```

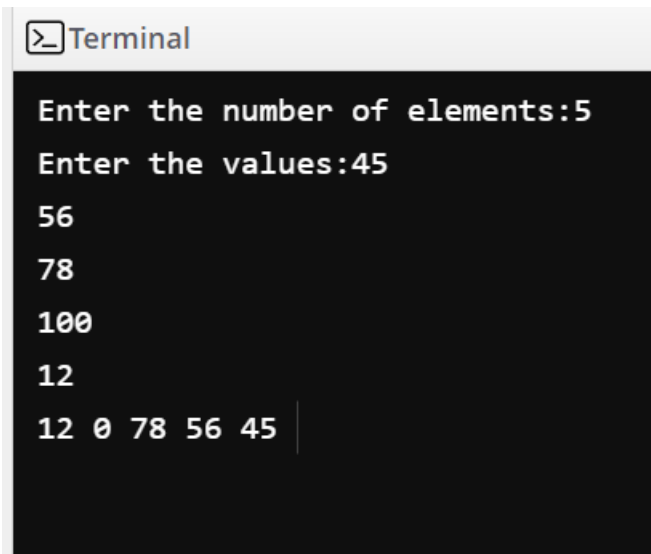
3. Insertion into unordered list, deletion of largest element in list

```
#include <stdio.h>
#include <stdlib.h>
struct node {
    int val;
    struct node* next;
};
struct node* newNode(int val) {
    struct node* temp = (struct node*) malloc(sizeof(struct node));
    temp->val = val;
    temp->next = NULL;
    return temp;
}
struct node* insert(struct node* head, int val) {
    struct node* new = newNode(val);
    new->next = head;
    return new;
}
struct node* deleteLargest(struct node* head) {
    struct node* temp = head;
    struct node* prev = NULL;
    struct node* largest = head;
    int max = head->val;
    while (temp != NULL) {
        if (temp->val > max) {
            max = temp->val;
            largest = temp;
            prev = temp;
        }
        temp = temp->next;
    }
    if (prev == NULL) {
        head = largest->next;
    }
    else {
        prev->next = largest->next;
    }
    free(largest);
    return head;
}
int main() {
    struct node* head = NULL;
    int n,i,val;
    printf("Enter the number of elements:");
    scanf("%d",&n);
    printf("Enter the values:");
    for(i=0;i<n;i++){
        scanf("%d",&val);
```

```

        head = insert(head, val);
    }
    head = deleteLargest(head);
    struct node* temp = head;
    while (temp != NULL) {
        printf("%d ", temp->val);
        temp = temp->next;
    }
    return 0;
}

```



A terminal window titled "Terminal" with a dark background. It shows the following text:

```

Enter the number of elements:5
Enter the values:45
56
78
100
12
12 0 78 56 45

```

In this program, user is asked to enter the number of elements and the values of those elements. With the help of insert() function, the entered values are added to the linked list. After that largest element is deleted using deleteLargest() function. Finally, the modified list is printed on the screen.