Nanduri Jayant Vishnu 20BCB0040

DSA MTT

Write a program to insert an element and to count and display the number of positive and

negative numbers in a singly linked list.

**CODE**  
  
*#include* <iostream>

using namespace std;

int i, j, choice;

struct Node

{

    int data;

    struct Node \*next;

};

struct Node \*head = NULL;

void disp()

{

    struct Node \*p;

    p = head;

*if* (p == NULL)

    {

        cout << "No element present" << endl;

    }

*else*

    {

        cout << "\*\*Linked List Elements\*\*" << endl;

*while* (p != NULL)

        {

            cout << p->data << endl;

            p = p->next;

        }

    }

}

void count\_positive\_negative()

{

    int positive = 0;

    int negative = 0;

    struct Node \*p;

    p = head;

*if* (head == NULL)

    {

        cout << "Linked List is empty" << endl;

    }

*else*

    {

*while* (p != NULL)

        {

*if* (p->data >= 0)

            {

                positive += 1;

            }

*else*

            {

                negative += 1;

            }

            p = p->next;

        }

    }

    cout << "Num of Positive elements: " << positive << endl;

    cout << "Num of Negative elements: " << negative << endl;

}

void insertAtBegin()

{

    struct Node \*temp = new (struct Node);

    int num;

    cout << "Enter data to be inserted: " << endl;

    cin >> num;

    temp->data = num;

*if* (head == NULL)

    {

        temp->next = NULL;

        head = temp;

        cout << "Node Inserted at Beginning" << endl;

    }

*else*

    {

        temp->next = head;

        head = temp;

        cout << "Node Inserted at Beginning" << endl;

    }

}

void insertAtEnd()

{

    struct Node \*temp, \*r;

    temp = new (struct Node);

    r = head;

    int num;

    cout << "Enter data to be inserted: " << endl;

    cin >> num;

    temp->data = num;

*if* (head == NULL)

    {

        temp->next = NULL;

        head = temp;

        cout << "Node Inserted at Ending" << endl;

    }

*else*

    {

*while* (r->next != NULL)

        {

            r = r->next;

        }

        temp->next = NULL;

        r->next = temp;

        cout << "Node Inserted at Ending" << endl;

    }

}

int main()

{

    cout << "1. Display the List" << endl;

    cout << "2. Insert at Starting" << endl;

    cout << "3. Insert at End" << endl;

    cout << "4. Count positive and negative element " << endl;

    cout << "5. Exit" << endl;

*while* (choice != 5)

    {

        cout << "Enter your choice" << endl;

        cin >> choice;

*switch* (choice)

        {

*case* 1:

            disp();

*break*;

*case* 2:

            insertAtBegin();

*break*;

*case* 3:

            insertAtEnd();

*break*;

*case* 4:

            count\_positive\_negative();

*break*;

*case* 5:

            cout << "Exiting..." << endl;

*break*;

*default*:

            cout << "Invalid input" << endl;

        }

    }

*return* 0;

}

**OUTPUT**  
  
  
  
  
1. Display the List

2. Insert at Starting

3. Insert at End

4. Count positive and negative element

5. Exit

Enter your choice

2

Enter data to be inserted:

17

Node Inserted at Beginning

Enter your choice

2

Enter data to be inserted:

177

Node Inserted at Beginning

Enter your choice

3

Enter data to be inserted:

-55

Node Inserted at Ending

Enter your choice

3

Enter data to be inserted:

2

Node Inserted at Ending

Enter your choice

2

Enter data to be inserted:

-55

Node Inserted at Beginning

Enter your choice

34

Invalid input

Enter your choice

3

Enter data to be inserted:

-88

Node Inserted at Ending

Enter your choice

1

\*\*Linked List Elements\*\*

-55

177

17

-55

2

-88

Enter your choice

4

Num of Positive elements: 3

Num of Negative elements: 3

Enter your choice

2

Enter data to be inserted:

5

Node Inserted at Beginning

Enter your choice

4

Num of Positive elements: 4

Num of Negative elements: 3

Enter your choice

5

Exiting...  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**CODE**  
  
*#include* <stdio.h>

void selection\_sort(int l[], int n)

{

    int minimum\_index;

*for* (int i = 0; i < n; i++)

    {

*for* (int i = 0; i < n; i++)

        {

            printf("%d ", l[i]);

        }

        printf("\n");

        minimum\_index = i;

*for* (int j = i + 1; j < n; j++)

        {

*if* (l[j] < l[minimum\_index])

            {

                minimum\_index = j;

            }

        }

        int temp;

        temp = l[i];

        l[i] = l[minimum\_index];

        l[minimum\_index] = temp;

    }

}

void main()

{

    int n;

    printf("Number of elements ? ");

    scanf("%d", &n);

    int arr[n];

    printf("Enter array elements:\n");

*for* (int i = 0; i < n; i++)

    {

        scanf("%d", &arr[i]);

    }

    printf("Iterations are:\n");

    selection\_sort(arr, n);

    printf("Sorted array:\n");

*for* (int i = 0; i < n; i++)

    {

        printf("%d ", arr[i]);

    }

}

**OUTPUT**  
  
  
  
  
Number of elements5

Enter array elements:

7

4

2

9

1

Iterations are:

7 4 2 9 1

1 4 2 9 7

1 2 4 9 7

1 2 4 9 7

1 2 4 7 9

Sorted array:

1 2 4 7 9