#include <stdio.h>

#include <stdlib.h>

struct node {

int data;

struct node \*next;

};

struct node \*start = NULL;

void insert\_at\_begin(int);

void insert\_at\_end(int);

void traverse();

void delete\_from\_begin();

void delete\_from\_end();

int count = 0;

int main () {

int i, data;

for (;;) {

printf("1. Insert an element at the beginning of linked list.\n");

printf("2. Insert an element at the end of linked list.\n");

printf("3. Traverse linked list.\n");

printf("4. Delete an element from beginning.\n");

printf("5. Delete an element from end.\n");

printf("6. Exit\n");

scanf("%d", &i);

if (i == 1) {

printf("Enter value of element\n");

scanf("%d", &data);

insert\_at\_begin(data);

}

else if (i == 2) {

printf("Enter value of element\n");

scanf("%d", &data);

insert\_at\_end(data);

}

else if (i == 3)

traverse();

else if (i == 4)

delete\_from\_begin();

else if (i == 5)

delete\_from\_end();

else if (i == 6)

break;

else

printf("Please enter valid input.\n");

}

return 0;

}

void insert\_at\_begin(int x) {

struct node \*t;

t = (struct node\*)malloc(sizeof(struct node));

t->data = x;

count++;

if (start == NULL) {

start = t;

start->next = NULL;

return;

}

t->next = start;

start = t;

}

void insert\_at\_end(int x) {

struct node \*t, \*temp;

t = (struct node\*)malloc(sizeof(struct node));

t->data = x;

count++;

if (start == NULL) {

start = t;

start->next = NULL;

return;

}

temp = start;

while (temp->next != NULL)

temp = temp->next;

temp->next = t;

t->next = NULL;

}

void traverse() {

struct node \*t;

t = start;

if (t == NULL) {

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

return;

}

printf("There are %d elements in linked list.\n", count);

while (t->next != NULL) {

printf("%d\n", t->data);

t = t->next;

}

printf("%d\n", t->data); // Print last node

}

void delete\_from\_begin() {

struct node \*t;

int n;

if (start == NULL) {

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

return;

}

n = start->data;

t = start->next;

free(start);

start = t;

count--;

printf("%d deleted from the beginning successfully.\n", n);

}

void delete\_from\_end() {

struct node \*t, \*u;

int n;

if (start == NULL) {

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

return;

}

count--;

if (start->next == NULL) {

n = start->data;

free(start);

start = NULL;

printf("%d deleted from end successfully.\n", n);

return;

}

t = start;

while (t->next != NULL) {

u = t;

t = t->next;

}

n = t->data;

u->next = NULL;

free(t);

printf("%d deleted from end successfully.\n", n);

}

