```c
#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;
  }

```

```c
  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;
```

```
133
134   if (start == NULL) {
135     printf("Linked list is empty.\n");
136     return;
137   }
138
139   count--;
140
141   if (start->next == NULL) {
142     n = start->data;
143     free(start);
144     start = NULL;
145     printf("%d deleted from end successfully.\n", n);
146     return;
147   }
148
149   t = start;
150
151   while (t->next != NULL) {
152     u = t;
153     t = t->next;
154   }
155
156   n = t->data;
157   u->next = NULL;
158   free(t);
159
160   printf("%d deleted from end successfully.\n", n);
161 }
```