```
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node* next;
};

struct Node* head = NULL;

void push(struct Node** head_ref, int new_data) {
   struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
   new_node->data = new_data;
   new_node->next = (*head_ref);
   (*head_ref) = new_node;
}

void pop() {
   struct Node* ptr;
   if (head == NULL) {
      printf("\nList is empty");
   } else {
      ptr = head;
      head = ptr->next;
      free(ptr);
      printf("\nNode deleted from the beginning...");
   }
}

void enqueue(int item) {
   struct Node* ptr, *temp;
```

```
    ptr = (struct Node*)malloc(sizeof(struct Node));
    ptr->data = item;
    ptr->next = NULL;
    if (head == NULL) {
        head = ptr;
        printf("\nNode inserted");
    } else {
        temp = head;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = ptr;
        printf("\nNode inserted");
    }
}

void dequeue() {
    struct Node* ptr;
    if (head == NULL) {
        printf("\nList is empty");
    } else {
        ptr = head;
        head = ptr->next;
        free(ptr);
        printf("\nNode deleted from the beginning...");
    }
}

int main() {
    // Sample input
    push(&head, 1);
```

```
push(&head, 2);
push(&head, 3);

printf("\nInitial linked list: ");
struct Node* current = head;
while (current != NULL) {
   printf("%d ", current->data);
   current = current->next;
}

pop();
printf("\nLinked list after pop operation: ");
current = head;
while (current != NULL) {
   printf("%d ", current->data);
   current = current->next;
}

enqueue(4);
printf("\nLinked list after enqueue operation: ");
current = head;
while (current != NULL) {
   printf("%d ", current->data);
   current = current->next;
}

dequeue();
printf("\nLinked list after dequeue operation: ");
current = head;
while (current != NULL) {
   printf("%d ", current->data);
```

```
        current = current->next;
    }
    return 0;
}
```

```
Initial linked list: 3 2 1
Node deleted from the beginning...
Linked list after pop operation: 2 1
Node inserted
Linked list after enqueue operation: 2 1 4
Node deleted from the beginning...
Linked list after dequeue operation: 1 4
```