|  |  |
| --- | --- |
| Singly linked list | Doubly linked list |
| The complexity of insertation and deletion is O(n) | The complexity of insertion and dletion is O(1) |
| It allows one-way traversal of elements. | It allows for two way traversal |
| It often use to implement stacks | It can be used to implement binary tree,heaps. |

Code:

#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node \*next;

};

struct Node\* createNode(int data) {

struct Node newNode = (struct Node) malloc(sizeof(struct Node));

if (newNode == NULL) {

printf("Memory allocation failed.\n");

exit(1);

}

newNode->data = data;

newNode->next = NULL;

return newNode;

}

void insertAtBeginning(struct Node \*\*head, int data) {

struct Node \*newNode = createNode(data);

newNode->next = \*head;

\*head = newNode;

}

void printList(struct Node \*head) {

struct Node \*temp = head;

while (temp != NULL) {

printf("%d -> ", temp->data);

temp = temp->next;

}

printf("NULL\n");

}

int main() {

struct Node \*head = NULL;

insertAtBeginning(&head, 3);

insertAtBeginning(&head, 2);

insertAtBeginning(&head, m);

printf("Linked List: ");

printList(head);

return 0;

}