# **Name: Abdurrahman Qureshi**

# **Roll No: 210451**

Practical No: “X”

Aim: Insertion in a Link List

**CODE:**

#include <stdio.h>

#include <stdlib.h>

struct node{

int val;

struct node \*next;

};

void print\_list(struct node \*head){

printf("H->");

while (head){

printf("%d->", head->val);

head = head->next;

}

printf("……\n\n");

}

void insert\_front(struct node \*\*head, int value){

struct node \*new\_node = NULL;

new\_node = (struct node \*)malloc(sizeof(struct node));

if (new\_node == NULL){

printf(" Out of memory");

}

new\_node->val = value;

new\_node->next = \*head;

\*head = new\_node;

}

void insert\_end(struct node \*\*head, int value){

struct node \*new\_node = NULL;

struct node \*last = NULL;

new\_node = (struct node \*)malloc(sizeof(struct node));

if (new\_node == NULL){

printf(" Out of memory");

}

new\_node->val = value;

new\_node->next = NULL;

if (\*head == NULL){

\*head = new\_node;

return;

}

last = \*head;

while (last->next)

last = last->next;

last->next = new\_node;

}

void insert\_after(struct node \*head, int value, int after){

struct node \*new\_node = NULL;

struct node \*tmp = head;

while (tmp){

if (tmp->val == after)

{ /\*found the node\*/

new\_node = (struct node \*)malloc(sizeof(struct node));

if (new\_node == NULL){

printf("Out of memory");

}

new\_node->val = value;

new\_node->next = tmp->next;

tmp->next = new\_node;

return;

}

tmp = tmp->next;

}

}

void insert\_before(struct node \*\*head, int value, int before){

struct node \*new\_node = NULL;

struct node \*tmp = \*head;

new\_node = (struct node \*)malloc(sizeof(struct node));

if (new\_node == NULL){

printf("Out of memory");

return;

}

new\_node->val = value;

if ((\*head)->val == before){

new\_node->next = \*head;

\*head = new\_node;

return;

}

while (tmp && tmp->next){

if (tmp->next->val == before){

new\_node->next = tmp->next;

tmp->next = new\_node;

return;

}

tmp = tmp->next;

}

/\*Before node not found\*/

free(new\_node);

}

void main(){

int count = 0, i, val, after, before;

struct node \*head = NULL;

printf("Enter no: of elements: ");

scanf("%d", &count);

for (i = 0; i < count; i++){

printf("Enter %dth element: ", i);

scanf("%d", &val);

insert\_front(&head, val);

}

printf("starting list: ");

print\_list(head);

printf("enter front element: ");

scanf("%d", &val);

insert\_front(&head, val);

printf("items after insertion: ");

print\_list(head);

printf("enter last element: ");

scanf("%d", &val);

insert\_end(&head, val);

printf("items after insertion: ");

print\_list(head);

printf("Enter an ele to insert in the list: ");

scanf("%d", &val);

printf("Insert after: ");

scanf("%d", &after);

insert\_after(head, val, after);

printf("List after insertion: ");

print\_list(head);

printf("Enter an ele to insert in the list: ");

scanf("%d", &val);

printf("Insert before: ");

scanf("%d", &before);

insert\_before(&head, val, before);

printf("List after insertion: ");

print\_list(head);}

**OUTPUT:**

