**Print Middle Most Node of a Linked List**

#include <stdio.h>

#include <stdlib.h>

struct node

{

int num;

struct node \*next;

};

void create(struct node \*\*);

void middlenode(struct node \*);

void release(struct node \*\*);

int main()

{

struct node \*p = NULL;

printf("Enter data into the list\n");

create(&p);

middlenode(p);

release (&p);

return 0;

}

void middlenode(struct node \*head)

{

struct node \*p, \*q;

int flag = 0;

q = p = head;

/\*for every two hops of q, one hop for p\*/

while (q->next != NULL)

{

q = q->next;

if (flag)

{

p = p->next;

}

flag = !flag;

}

if (flag)

{

printf("List contains even number of nodes\nThe middle two node's values are: %d %d\n", p->next->num, p->num);

}

else

{

printf("The middle node of the list is: %d\n", p->num);

}

}

void create(struct node \*\*head)

{

int c, ch;

struct node \*temp;

do

{

printf("Enter number: ");

scanf("%d", &c);

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

temp->num = c;

temp->next = \*head;

\*head = temp;

printf("Do you wish to continue [1/0]: ");

scanf("%d", &ch);

} while (ch != 0);

printf("\n");

}

void release(struct node \*\*head)

{

struct node \*temp = \*head;

\*head = (\*head)->next;

while ((\*head) != NULL)

{

free(temp);

temp = \*head;

(\*head) = (\*head)->next;

}

}