Linked list

#include <stdio.h>

#include <stdlib.h>

struct node{

int info;

struct node \*ptr;

}\*top,\*top1,\*temp;

int topelement();

void push(int data);

void pop();

void empty();

void display();

void destroy();

void stack\_count();

void create();

int count = 0;

int main(){

int no, ch, e;

printf("\n 1 - Push");

printf("\n 2 - Pop");

printf("\n 3 - display");

printf("\n 4 - exit");

create();

while (1){

printf("\n Enter choice : ");

scanf("%d", &ch);

switch (ch){

case 1:

printf("Enter element : ");

scanf("%d", &no);

push(no);

break;

case 2:

pop();

break;

case 4:

exit(0);

case 3:

display();

break;

default :

printf(" wrong choice:Try again ");

break;

}

}

}

void create(){

top = NULL;

}

void push(int data){

if (top == NULL){

top =(struct node \*)malloc(1\*sizeof(struct node));

top->ptr = NULL;

top->info = data;

}

else{

temp =(struct node \*)malloc(1\*sizeof(struct node));

temp->ptr = top;

temp->info = data;

top = temp;

}

count++;

}

void display(){

top1 = top;

if (top1 == NULL){

printf("empty stack");

return;

}

while (top1 != NULL){

printf("%d ", top1->info);

top1 = top1->ptr;

}

}

void pop(){

top1 = top;

if (top1 == NULL){

printf("\n error");

return;

}

else

top1 = top1->ptr;

printf("\n Popped value : %d", top->info);

free(top);

top = top1;

count--;

}

int topelement(){

return(top->info);

}

Output:

