**DISPLAYING STACK FROM BOTTOM TO TOP USING RECURSIVE FUNCTION**

#include<stdio.h>

#include<stdlib.h>

struct Node {

int data;

struct Node \*link;

};

int isEmpty(struct Node\* top) {

if(top==NULL)

return 1;

else

return 0;

}

struct Node\* push(struct Node \*top, int data) {

struct Node\* ptr = (struct Node\*)malloc(sizeof(struct Node));

ptr->data = data;

ptr->link= top;

top = ptr;

return top;

}

}

int pop(struct Node\*\* top) {

if(isEmpty(\*top)) {

printf("Stack is empty! No element can be poppped!!");

}

else {

struct Node\* temp= \*top;

int x= temp->data;

\*top = (\*top) -> link;

free(temp);

return x;

}

}

void display(struct Node\* ptr) {

while(ptr!=NULL) {

printf("%d\t", ptr->data);

ptr=ptr->link;

}

}

void PrintReverseStack(struct Node\* temp) {

if(isEmpty(temp)) {

return ;

}

int x = pop(&temp);

PrintReverseStack(temp);

printf("%d\t",x);

temp=push(temp,x);

}

int main() {

struct Node\* top=NULL;

int n, choice;

while(1){

printf (" MENU ");

printf("\n1.Push element to stack\t\t2.Pop element from stack\n");

printf("3.Display the stack\t\t\t4.Peek\t\t\t5.Display stack in reverse order\t\t\t6.Exit\n");

printf("Enter your choice : ");

scanf("%d",&choice);

switch(choice){

case 1 : printf("\nEnter element : ");

scanf("%d",&n);

top = push(top,n);

break;

case 2 : printf("\n");

int num = pop(&top);

printf("\nDeleted %d from stack\n",num);

break;

case 3 : printf("\nDisplaying the stack ...\n");

display(top);

break;

case 4 : printf("\nDisplaying the top of the stack...");

printf("%d ",peek(top));

break;

case 5 : printf("\n");

PrintReverseStack(top);

break;

case 6 : exit(0);

default: printf("\nINVALID CHOICE !!!\n");

}

} return 0;

}