**Implementation of stack using linked list:**

**//**code

#include <stdio.h>

#include <stdlib.h>

// Implementation of stack using linked list;

// declaration of linked list

struct node {

int data;

struct node \*next;

};

// declaration of top;

struct node \*TOP = NULL;

void push(int val) { // push value on to stack

// declaring node and allocating memory

struct node \*newNode;

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

// inserting value in the node

newNode->data = val;

// Linking nodes of the list

newNode->next = TOP;

// shifting TOP

TOP = newNode;

}

void pop() { // pop element from the stack

// declaring traversing ptr

struct node \*ptr;

ptr = TOP;

if (TOP == NULL) { // Check if the stack is empty

printf("\nStack is empty");

return;

}

// printing the top element

printf("\nPopped element is : %d", TOP->data);

// shifting top to second element

TOP = ptr->next;

// deleting node from the memory

free(ptr);

}

void peek() { // prints top element from the stack

if (TOP == NULL) { // checks if stack is empty

printf("\nStack is empty!");

return;

}

// prints the top most element

printf("\nTop element of stack is : %d", TOP->data);

}

int size() {

// declaring a traversing pointer

struct node \*ptr;

ptr = TOP;

int count = 1;

if (TOP == NULL) { // if the stack is empty

return 0;

}

while (ptr->next != NULL) { // traverse the stack and increase the counter

ptr = ptr->next;

count++;

}

return count;

}

void display() { // display the complete stack

// declaring traversing pointer

struct node \*ptr;

ptr = TOP;

if (TOP == NULL) { // check if the list is empty

printf("\nStack is empty!");

return;

}

printf("\nElements in the stack are : ");

while (ptr->next != NULL) { // traverse while printing

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

ptr = ptr->next;

}

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

}

int main() {

int choice, val;

while (1) {

printf("\n\*1. PUSH");

printf("\n\*2. POP");

printf("\n\*3. PEEK");

printf("\n\*4. SIZE");

printf("\n\*5. DISPLAY");

printf("\n\*6. EXIT");

printf("\nEnter your choice : ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("\nEnter an element to push : ");

scanf("%d", &val);

push(val);

break;

case 2:

pop();

break;

case 3:

peek();

break;

case 4:

printf("\nSize of stack is : %d", size());

break;

case 5:

display();

break;

case 6:

printf("\n \*\*\* E X I T I N G \*\*\*");

exit(1);

default:

printf("\nINVALID INPUT");

}

}

return 0;

}

// output





