|  |
| --- |
| //Write a C Program to implement various operations of Singly Linked List Stack |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | #include <stdio.h> |
|  | #include <stdlib.h> |
|  | #define TRUE 1 |
|  | #define FALSE 0 |
|  |  |
|  | struct node |
|  | { |
|  | int data; |
|  | struct node \*next; |
|  | }; |
|  | typedef struct node node; |
|  |  |
|  | node \*top; |
|  |  |
|  | void initialize() |
|  | { |
|  | top = NULL; |
|  | } |
|  |  |
|  | void push(int value) |
|  | { |
|  | node \*tmp; |
|  | tmp = malloc(sizeof(node)); |
|  | tmp -> data = value; |
|  | tmp -> next = top; |
|  | top = tmp; |
|  | } |
|  |  |
|  | int pop() |
|  | { |
|  | node \*tmp; |
|  | int n; |
|  | tmp = top; |
|  | n = tmp->data; |
|  | top = top->next; |
|  | free(tmp); |
|  | return n; |
|  | } |
|  |  |
|  | int Top() |
|  | { |
|  | return top->data; |
|  | } |
|  |  |
|  | int isempty() |
|  | { |
|  | return top==NULL; |
|  | } |
|  |  |
|  | void display(node \*head) |
|  | { |
|  | if(head == NULL) |
|  | { |
|  | printf("NULL\n"); |
|  | } |
|  | else |
|  | { |
|  | printf("%d\n", head -> data); |
|  | display(head->next); |
|  | } |
|  | } |
|  |  |
|  | int main() |
|  | { |
|  | initialize(); |
|  | push(10); |
|  | push(20); |
|  | push(30); |
|  | printf("The top is %d\n",Top()); |
|  | pop(); |
|  | printf("The top after pop is %d\n",Top()); |
|  | display(top); |
|  | return 0; |
|  | } |