

```

1  //Stack implementation using Linked List
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <conio.h>
5  #include <malloc.h>
6  struct stack
7  {
8      int data;
9      struct stack *next;
10 };
11 struct stack *top = NULL;
12 struct stack *push(struct stack *, int);
13 struct stack *display(struct stack *);
14 struct stack *pop(struct stack *);
15 int peek(struct stack *);
16 int main()
17 {
18     int val, option;
19     do
20     {
21         printf("\n *****MAIN MENU*****");
22         printf("\n 1. PUSH");
23         printf("\n 2. POP");
24         printf("\n 3. PEEK");
25         printf("\n 4. DISPLAY");
26         printf("\n 5. EXIT");
27         printf("\n Enter your option: ");
28         scanf("%d", &option);
29         switch(option)
30         {
31             case 1:
32                 printf("\n Enter the number to be pushed on stack: ");
33                 scanf("%d", &val);
34                 top = push(top, val);
35                 break;
36             case 2:
37                 top = pop(top);
38                 break;
39             case 3:
40                 val = peek(top);
41                 if (val != -1)
42                     printf("\n The value at the top of stack is: %d", val);
43                 else
44                     printf("\n STACK IS EMPTY");
45                 break;
46             case 4:
47                 top = display(top);
48                 break;
49         }
50     }while(option != 5);
51     return 0;
52 }
53 struct stack *push(struct stack *top, int val)
54 {
55     struct stack *ptr;
56     ptr = (struct stack*)malloc(sizeof(struct stack));
57     ptr->data = val;
58     if(top == NULL)
59     {
60         ptr->next = NULL;
61         top = ptr;
62     }
63     else
64     {
65         ptr->next = top;
66         top = ptr;
67     }
68     return top;
69 }

```

```

70  struct stack *display(struct stack *top)
71  {
72      struct stack *ptr;
73      ptr = top;
74      if(top == NULL)
75          printf("\n STACK IS EMPTY");
76      else
77      {
78          while(ptr != NULL)
79          {
80              printf("\n %d", ptr -> data);
81              ptr = ptr -> next;
82          }
83      }
84      return top;
85  }
86  struct stack *pop(struct stack *top)
87  {
88      struct stack *ptr;
89      ptr = top;
90      if(top == NULL)
91          printf("\n STACK UNDERFLOW");
92      else
93      {
94          top = top -> next;
95          printf("\n The value being deleted is: %d", ptr -> data);
96          free(ptr);
97      }
98      return top;
99  }
100  int peek(struct stack *top)
101  {
102      if(top==NULL)
103          return -1;
104      else
105          return top ->data;
106  }
107

```