

exp_6/a.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct Node {
5     int data;
6     struct Node* next;
7 };
8
9 struct Stack {
10    struct Node* top;
11 };
12
13 void push(struct Stack* stack, int data) {
14     struct Node* new_node = (struct Node*) malloc(sizeof(struct Node));
15     new_node->data = data;
16     new_node->next = stack->top;
17     stack->top = new_node;
18 }
19
20 int pop(struct Stack* stack) {
21     if (stack->top == NULL) return -1;
22     struct Node* temp = stack->top;
23     int popped = temp->data;
24     stack->top = temp->next;
25     free(temp);
26     return popped;
27 }
28
29 int peek(struct Stack* stack) {
30     if (stack->top == NULL) return -1;
31     return stack->top->data;
32 }
33
34 int main() {
35     struct Stack stack;
36     stack.top = NULL;
37     int choice, value;
38
39     while (1) {
40         printf("1: Push, 2: Pop, 3: Peek, 0: Exit\nEnter choice: ");
41         scanf("%d", &choice);
42         if (choice == 0) break;
43         switch (choice) {
44             case 1:
45                 printf("Enter value to push: ");
46                 scanf("%d", &value);
47                 push(&stack, value);
48                 break;
49         }
50     }
51 }
```

```
49     case 2:
50         value = pop(&stack);
51         if (value == -1) printf("Stack Empty\n");
52         else printf("Popped %d\n", value);
53         break;
54     case 3:
55         value = peek(&stack);
56         if (value == -1) printf("Stack Empty\n");
57         else printf("Top element: %d\n", value);
58         break;
59     default:
60         printf("Invalid choice\n");
61     }
62 }
63 return 0;
64 }
65 }
```