```c
#include <stdio.h>
#include <stdlib.h>

// Define a structure for a stack node
struct stack
{
    int data;            // Data stored in the node
    struct stack *next;  // Pointer to the next node in the stack
};

// Function prototypes
struct stack *push(struct stack *, int);
struct stack *pop(struct stack *);
int peek(struct stack *);
struct stack *display(struct stack *);

int main() {
    struct stack *top = NULL;  // Initialize the stack as empty
    int val, option;

    do {
        printf("\n *****MAIN MENU*****");
        printf("\n 1. PUSH");
        printf("\n 2. POP");
        printf("\n 3. PEEK");
        printf("\n 4. DISPLAY");
        printf("\n 5. EXIT");
        printf("\n Enter your option: ");
        scanf("%d", &option);

        switch (option) {
            case 1:
                printf("\n Enter the number to be pushed on stack: ");
                scanf("%d", &val);
                top = push(top, val);
                break;
            case 2:
                top = pop(top);
                break;
            case 3:
                val = peek(top);
                if (val != -1)
                    printf("\n The value at the top of stack is: %d", val);
                else
                    printf("\n STACK IS EMPTY");
```

```
                break;
            case 4:
                top = display(top);
                break;
        }
    } while (option != 5);

    return 0;
}

// Function to push a value onto the stack
struct stack *push(struct stack *top, int val) {
    struct stack *ptr;
    ptr = (struct stack *)malloc(sizeof(struct stack));
    ptr->data = val;

    if (top == NULL) {
        ptr->next = NULL;
        top = ptr;
    } else {
        ptr->next = top;
        top = ptr;
    }
    return top;
}

// Function to pop a value from the stack
struct stack *pop(struct stack *top) {
    struct stack *ptr;
    ptr = top;

    if (top == NULL) {
        printf("\n STACK UNDERFLOW");
    } else {
        top = top->next;
        printf("\n The value being deleted is: %d", ptr->data);
        free(ptr);
    }
    return top;
}

// Function to peek at the top value of the stack
int peek(struct stack *top) {
    if (top == NULL)
        return -1;
```

```
    else
        return top->data;
}

// Function to display the elements of the stack
struct stack *display(struct stack *top) {
    struct stack *ptr;
    ptr = top;

    if (top == NULL)
        printf("\n STACK IS EMPTY");
    else {
        printf("\n Elements of the stack: ");
        while (ptr != NULL) {
            printf("%d ", ptr->data);
            ptr = ptr->next;
        }
    }
    return top;
}
```