stack-using-linkedlist.c

```
//write a program to implement stack using a linked list.

#include "linkedlist.h"
//operatioins of linked list are in the header file made by me. visit: https://github.com/devSusanta/DSA/blob/main/linkedlist.h

void main(){
   lkdlist *stack1 = NULL;
   int op,x;
   while(1){
      printf("1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit.\nEnter Your choice: ");
      scanf("%d",&op);
      switch(op){
         case 1:
            printf("Enter the item: ");
            scanf("%d",&x);
            push_top(&stack1,x);
            break;
         case 2:
            if(isEmpty(stack1)){
               printf("Underflow.\n");
            }else{
               printf("Removed %d from stack\n",pop_top(&stack1));
            }
            break;
         case 3:
            if(isEmpty(stack1)){
               printf("Stack is empty.\n");
            }else{
               printf("The elements of stack are: ");
               display(stack1);
            }
            break;
         case 4:
            printf("%d\n",len(stack1));
            break;
         case 5:
            if(isEmpty(stack1)){
               printf("Stack is Empty\n");
            }else{
               printf("Stack is Not Empty\n");
            }
            break;
         case 6:
            printf("Don't worry stack is never going to be full, you have enough space remain.\n");
            break;
         case 7:
            if(isEmpty(stack1)){
               printf("Stack is Empty\n");
            }else{
               printf("%d\n",top(stack1));
            }
            break;
         case 8:
            if(isEmpty(stack1)){
               printf("Stack is Empty\n");
            }else{
               printf("%d\n",bottom(stack1));
            }
            break;
         case 9: exit(0);
         default: printf("Invalid Input.\n");
```

```
    }
  }
```

}OUTPUT

```
PS S:\WorkSpace\CollegeWork\DataStructure> gcc .\stack-using-linked-list.c
PS S:\WorkSpace\CollegeWork\DataStructure> ./a
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit.
Enter Your choice: 1 12 1 13 1 14 1 15 1 16 1 18 1 109
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit.
Enter Your choice: 3
The elements of stack are: 109 -> 18 -> 16 -> 15 -> 14 -> 13 -> 12 -> NULL
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit.
Enter Your choice: 2 2 2
Removed 109 from stack
Removed 18 from stack
Removed 16 from stack
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit.
Enter Your choice: 3
The elements of stack are: 15 -> 14 -> 13 -> 12 -> NULL
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit.
Enter Your choice: 4
4
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit.
Enter Your choice: 7
15
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit.
Enter Your choice: 8
12
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit.
Enter Your choice: 9
PS S:\WorkSpace\CollegeWork\DataStructure>
```